

DAS Departamento de Automação e Sistemas
CTC Centro Tecnológico
UFSC Universidade Federal de Santa Catarina

Identificação de Sistemas Dinâmicos Não Lineares Utilizando Métodos de Colocação

*Monografia submetida à Universidade Federal de Santa Catarina
como requisito para a aprovação da disciplina:*

DAS 5511: Projeto de Fim de Curso

Ricardo Santos da Silva

Florianópolis, agosto de 2014

Identificação de Sistemas Dinâmicos Não Lineares Utilizando Métodos de Colocação

Ricardo Santos da Silva

Esta monografia foi julgada no contexto da disciplina
DAS 5511: Projeto de Fim de Curso
e aprovada na sua forma final pelo
Curso de Engenharia de Controle e Automação Industrial

Banca Examinadora:

Marco Aurélio Schmitz de Aguiar
Orientador Empresa

Eduardo Camponogara
Orientador do Curso

Ricardo José Rabelo
Responsável pela disciplina

, Avaliador

, Debatedor

, Debatedor

Resumo

Este estudo busca o desenvolvimento de técnicas de identificação de modelos para sistemas de produção de petróleo utilizando o método de colocação (*collocation method*). Combinando este método com técnicas de otimização não-linear resolve-se um problema de encontrar os parâmetros e condições iniciais de um modelo dinâmico não linear que melhor descrevem a resposta de um sistema no tempo.

O *collocation method* é um caso particular do método Implícito de Runge-Kutta (IRK), que consiste na solução numérica de equações diferenciais e equações diferenciais algébricas.

Em particular, inicialmente verifica-se se a estratégia é realizável utilizando um simples circuito RLC e, após a validação, a técnica é utilizada para resolver a identificação de um poço produtor de petróleo e gás. Para identificação da resposta do sistema foi utilizado um modelo de baixa ordem, com apenas 3 estados, que pode ser utilizado numa malha de controle. Utilizando os dados de simulação deste modelo de baixa ordem, o problema de identificação é facilmente resolvido com boa precisão. Estes resultados iniciais impulsionaram a busca de dados reais ou de simuladores complexos de fluxo multifásico (e.g.; *OLGA*) para serem usados como referências da estratégia.

A implementação foi realizada com o emprego de ferramentas de modelagem como *Modelica* e *solvers* de otimização disponibilizados através do pacote *Optimica*.

Abstract

This study seeks to develop techniques for identifying models for oil production systems using the collocation method. Combining this method with nonlinear optimization techniques results in an effective strategy to find the parameters and initial conditions of a nonlinear dynamic model that best describe the time response of a system.

The collocation method is a particular case of the Implicit Runge-Kutta (IRK) method, which is suitable for the numerical solution of differential equations and differential algebraic equations.

After testing and validating the collocation method with a simple application to the identification of the parameters of a RLC circuit, using a simple RLC circuit and after validation, the technique is used to solve the identification of an oil and gas producing well. To identify the system response it was used a low-order model, with only three states, which can be used in a control loop. Using the simulation data of this low-order model, the identification problem is easily solved with good accuracy. The initial results have motivated the search for real data or complex multiphase flow simulators (eg: OLGA) to be used as reference to the strategy.

The implementation was carried out with the use of modeling tools like Modelica and the optimization solvers available through the Optimica package.

Contents

1	Introdução	1
2	<i>Collocation Method</i>	3
2.1	Sistema Dinâmicos	3
2.2	Filtro de Kalman	5
2.3	Collocation Method	8
2.4	Polinômio Deslocado de Gauss-Jacobi	12
2.5	Ordem de Convergência	14
2.6	Exemplo	14
2.6.1	Radau Collocation	14
2.6.2	Gauss Collocation	16
3	Ferramentas de Implementação	20
3.1	Modelica	20
3.2	Optimica	20
3.3	JModelica.org	22
3.3.1	CasADi	23
3.3.2	Sundials	23
3.3.3	IPOPT	23
4	Estimação de Parâmetros	25
4.1	Otimização	25
4.1.1	Função Objetivo	27
4.2	Circuito RLC	29
4.2.1	Estimação dos Parâmetros do Circuito RLC	31

4.3	Poço Produtor de Petróleo	36
4.3.1	Modelo do Poço	37
4.3.2	Estimação dos Parâmetros de um Poço Produtor de Petróleo . .	39
4.4	Estimação do poço com dados	42
4.4.1	Annulus	42
4.4.2	Tubing	42
5	Conclusão	47
	Bibliography	48
	Appendix A: Modelos Dinâmicos	50
A.1	Modelo do Poço	50
A.2	Annulus	53
A.3	Tubing II	55
	Appendix B: Modelos de Simulação	59
B.1	Modelo RLC	59
B.2	Modelo de Otimização do RLC	59
B.3	Código Python para Simulação e Estimação	60

Chapter 1: Introdução

Estimação é o processo de inferir o valor de uma quantidade de interesse a partir de observações indiretas, imprecisas e incertas. O propósito da observação pode ser, por exemplo: (a) a determinação da órbita de planetas (Laplace, Gauss, Legendre); (b) a determinação da posição e velocidade de uma aeronave em um sistema de controle de tráfego aéreo; (c) a determinação de parâmetros de um modelo de sistema como forma de prever o estado físico do mesmo, entre outros.

Em um processo real, existem diversos fatores que podem alterar seu comportamento. Tais fatores são facilmente notados ao modelar o processo: ao alterar os parâmetros verificamos uma mudança nas variáveis de saída. Por exemplo, a resistência nos fios de transmissão de energia influenciam a corrente de um circuito elétrico. Essas mudanças, mesmo que sutis, podem influenciar na performance e estabilidade de técnicas de controle.

Este trabalho é a extensão de um trabalho anterior [15], onde foram desenvolvidos modelos não lineares de uma plataforma de produção de petróleo e gás. Com o conhecimento do modelo e possuindo medições do processo é possível estimar os parâmetros dos modelos, com a utilização de uma ferramenta para amostragem e coleta de dados chamada *collocation method* [12], [9], e técnicas de otimização. Existem outras técnicas como o filtro de Kalman, *maximum likelihood*, entre outras que permitem a estimação de parâmetros. Entretanto, “verificou-se que as técnicas de otimização são mais robustas e viáveis do que o filtro de Kalman”[2]. “Para sistemas não-lineares, o filtro de Kalman é comumente aplicado na prática. Enquanto o filtro de Kalman é relativamente fácil de implementar, tem sido demonstrado que o filtro possui um desempenho insatisfatório em sistemas altamente não-lineares” [13].

As equações dos modelos foram codificadas usando uma linguagem de alto nível para modelagem de sistemas chamada Modelica, baseada em equações e orientada a objetos projetada para modelagem gráfica e textual de sistemas físicos complexos. O ambiente computacional JModelica.org foi usado para compilar o modelo. Este ambiente também faz a interface do modelo com ferramentas de simulação e de otimização.

O Capítulo 2 descreve o *collocation method*, ferramenta utilizada para a estimação dos parâmetros. O Capítulo 3 apresenta as ferramentas utilizadas para a

implementação dos problemas. O Capítulo 4 descreve os problemas de estimação e seus resultados. O Capítulo 5 apresenta as conclusões finais e possibilidades de trabalhos futuros.

Chapter 2: *Collocation Method*

Neste Capítulo será possível ver uma explicação sobre sistemas dinâmicos, os quais terão seus parâmetros estimados pelo *collocation method* (combinado de técnicas de otimização). Desta forma, este *collocation method* é um caso particular do método Implícito de Runge-Kutta (IRK), onde demonstra-se o equacionamento e dedução deste.

Além dessa explicação, uma outra forma de estimação de parâmetros também é descrita neste Capítulo: os filtros de Kalman. Este, porém, é citado brevemente, por conta das desvantagens que apresenta perante o *collocation method*.

Ao final do Capítulo, portanto, apresenta-se um exemplo do *collocation method*, por ser um método mais adequado neste trabalho, para melhor entendimento.

2.1: Sistema Dinâmicos

Um sistema dinâmico é um conceito matemático, caracterizado por uma regra fixa que descreve a dependência do tempo de um ponto em um espaço geométrico, portanto é um sistema que evolui no tempo. Normalmente, estes sistemas têm entradas e saídas. A qualquer momento, um sistema dinâmico tem um estado determinado por um vetor de números reais que pode ser representado por um ponto em um espaço de estado apropriado. Pequenas variações no estado do sistema criam pequenas mudanças nos valores do sistema. A regra de evolução do sistema dinâmico descreve o que os estados futuros seguem a partir do estado atual. A regra é determinística; em outras palavras, para um dado intervalo de tempo apenas um estado futuro decorre do estado atual. A regra normalmente é definida por um conjunto de equações diferenciais algébricas (DAE), no molde de

$$\frac{dx}{dt} = f(x(t), u(t), w(t), t) \quad (2.1a)$$

$$g(x(t), u(t), w(t), t) = 0 \quad (2.1b)$$

onde $x(t)$ é o vetor de estados do sistema, $u(t)$ é o vetor de sinais de controle, $w(t)$ é o vetor com as variáveis algébricas e f e g são funções que envolvem todos os componentes anteriores.

Uma vez que o sistema possa ser resolvido, dado um ponto inicial, é possível determinar todas as suas posições futuras, uma coleção de pontos conhecidos como trajetória. Para sistemas dinâmicos simples, conhecendo a trajetória é muitas vezes suficiente para obter suas posições futuras, mas a maioria dos sistemas dinâmicos são muito complicados para serem entendidos em termos de trajetórias individuais. As dificuldades surgem porque:

- Os sistemas estudados somente são conhecidos de maneira aproximada. Os parâmetros do sistema podem não ser conhecidos com precisão ou termos da equação podem estar faltando. As aproximações utilizadas podem invalidar a validade ou relevância de soluções numéricas. Para responder a estas perguntas várias noções de estabilidade foram introduzidas no estudo de sistemas dinâmicos, como a estabilidade de Lyapunov ou estabilidade estrutural. A estabilidade do sistema dinâmico implica que há uma classe de modelos ou condições iniciais para os quais as trajetórias sejam equivalentes. A operação para a comparação de órbitas para estabelecer sua equivalência muda com as diferentes noções de estabilidade.
- O tipo de trajetória pode ser mais importante do que uma trajetória particular. Algumas trajetórias podem ser periódicas, enquanto outras podem vagar por diferentes estados do sistema. A classificação de todas as trajetórias possíveis levou ao estudo qualitativo de sistemas dinâmicos, isto é, propriedades que não mudam com a alteração de coordenadas. Sistemas dinâmicos lineares são exemplos de sistemas dinâmicos em que as classes de possíveis órbitas são compreendidas.
- O comportamento de trajetórias como uma função de um parâmetro pode ser o que é necessário para uma aplicação. Como o parâmetro pode ser variado, os sistemas dinâmicos podem ter pontos de bifurcação, onde o comportamento qualitativo do sistema dinâmico muda. Uma bifurcação ocorre quando uma pequena mudança feita nos valores dos parâmetros de um sistema provoca uma mudança repentina em seu comportamento. O ponto onde essa mudança ocasiona a bifurcação é chamado de ponto de bifurcação. Por exemplo, pode ir de ter apenas movimentos periódicos ao comportamento aparentemente errático, como na transição para escoamento turbulento de um fluido. Bifurcações ocorrem em ambos os sistemas contínuos e sistemas discretos.

2.2: Filtro de Kalman

O filtro de Kalman é um estimador denominado problema linear quadrático, o qual consiste no problema de estimar o estado instantâneo de um sistema dinâmico linear perturbado por ruído branco, através do uso de medições linearmente relacionadas com este estado, porém corrompidas pelo ruído branco.

Uma das primeiras aplicações do filtro de Kalman foi o controle de sistemas dinâmicos complexos tais como: processos de manufatura, aeronáutica e astronáutica, navios, etc. Desde o momento da sua introdução, o filtro de Kalman tem sido objeto de extensa pesquisa e aplicação, em especial na área de navegação autônoma ou assistida. Isto é provavelmente devido, em grande parte, aos avanços da computação digital que tornaram prática a utilização do filtro, mas também à relativa simplicidade e robustez do próprio filtro. Raramente as condições necessárias para otimização realmente existem e, no entanto, o filtro funciona bem para muitas aplicações apesar desta situação [16].

Para controlar um sistema dinâmico, precisa-se primeiro saber o que ele está fazendo. Para estas aplicações, nem sempre é possível ou desejável medir toda variável que se deseja controlar, e o filtro de Kalman permite uma maneira de inferir a informação faltante através de medições indiretas e ruídos.

Sob o ponto de vista prático, algumas perspectivas do filtro de Kalman [7] podem ser consideradas:

- Trata-se apenas de uma ferramenta: Ele não resolve nenhum problema por si mesmo. Não se trata de uma ferramenta física, de um ente ou ferramenta matemática para tornar o trabalho mental mais eficiente.
- É um programa computacional: Ele foi chamado de idealmente preparado para um computador digital, em parte porque utiliza uma representação finita de um problema de estimação por um número fixo de variáveis. Entretanto, admite-se que essas variáveis sejam números reais, com precisão infinita.
- Trata-se de uma caracterização estatística de um problema de estimação: Ele é mais que um estimador, porque ele propaga o estado corrente de conhecimento de um sistema dinâmico, incluindo a influência estatística de perturbações randômicas e os efeitos de todas as medições passadas.

O filtro age de forma a minimizar o erro médio quadrático, que é a diferença entre o estado predito e o atual. Dado um valor inicial, o filtro prediz o próximo estado e, baseado na leitura do estado, atualiza a predição e minimiza o erro em cada atualização.

A característica recursiva do filtro proporciona uma solução computacionalmente aceitável, uma vez que a computação ocorre à medida que os dados vão sendo processados. O filtro apoia-se no fato de que o ruído é normalmente distribuído. Mesmo que essa hipótese não se confirme, o filtro continua sendo um estimador ótimo.

O filtro de Kalman é baseado em um modelo dinâmico de sistema que representa como seus parâmetros, representados por um vetor de estados, que variam à medida que o filtro é processado [11]. O modelo é representado por:

$$x(k) = F(k, k - 1)x(k - 1) + n(k) \quad (2.2)$$

onde $x(k)$ representa o vetor de estados, $F(k, k - 1)$ é uma matriz de transição de estados que mostra como o processo transita do estado $k - 1$ para o estado k , $n(k)$ é um ruído do sistema e k representa a iteração do processo ao longo do tempo.

Além do modelo dinâmico do sistema, o filtro tem como base a observação (medida) do sistema, retirado de sensores, que relaciona os sinais analíticos com o vetor de estado e é descrita por:

$$y(k) = h(k)x(k - 1) + e(k) \quad (2.3)$$

onde $y(k)$ representa o vetor de medição dos sinais analíticos, $h(k)$ é a função de medição que relaciona o vetor de estados com o sinal analítico e $e(k)$ é o ruído de medição.

O filtro de Kalman funciona em duas etapas: predição e atualização. Na fase de predição o estado atual é estimado com base na leitura realizada do estado anterior. Na fase de atualização o estado predito é atualizado com base na leitura do estado em tempo real. Para ajustar a diferença entre o vetor de estado predito e do atualizado é utilizado o ganho de Kalman, descrito por

$$g(k) = P(k - 1)h(k)[h(k)P(k - 1)h(k) + r(k)]^{-1} \quad (2.4)$$

onde $P(k)$ é a matriz de covariância do sistema e $r(k)$ é a variância do ruído de medição.

A situação ideal é quando o ganho de Kalman converge a zero à medida que o vetor de estados vai se estabilizando. Entretanto, na prática, o ganho de Kalman converge a um intervalo próximo de zero e é determinado pelo ruído do sistema. A matriz de covariância do sistema pode ser obtida como

$$P(k) = [I - g(k-1)\dot{h}(k)]P(k-1)[I - g(k-1)\dot{h}(k)] + g(k-1)r(k)\dot{g}(k-1) \quad (2.5)$$

onde I representa a matriz identidade.

Depois de encontrado o ganho de Kalman e atualizada a matriz de covariância através das Equações (2.4) e (2.5), respectivamente, é feita a atualização do vetor de estados por

$$x(k) = x(k-1) + g(k)[y(k) - \dot{h}(k)x(k-1)] \quad (2.6)$$

Um ponto crucial no filtro de Kalman é a estimativa dos valores iniciais do vetor de estados e da matriz de covariância do sistema, uma vez que no começo do processo não existem valores anteriores ao inicial no sinal. A Figura 2.1 apresenta o processo das etapas do filtro.

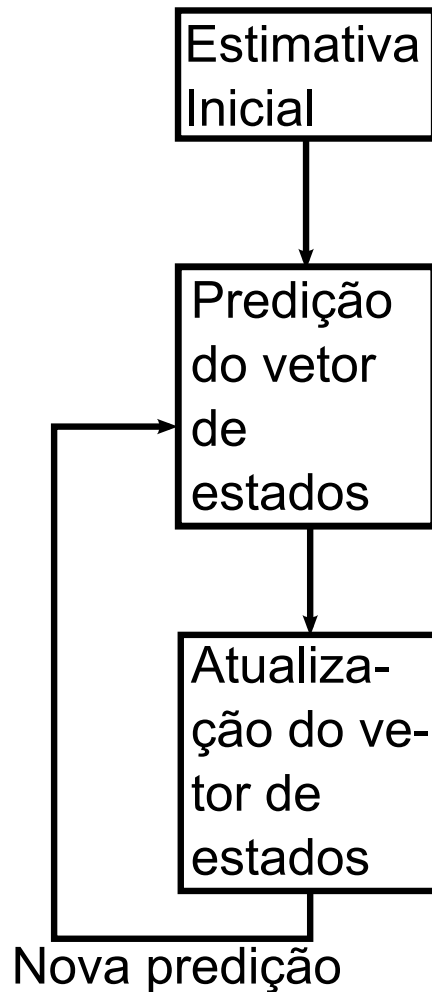


Figure 2.1: Filtro de Kalman

2.3: Collocation Method

O *collocation method* é um método para a solução numérica de equações diferenciais e equações diferenciais algébricas (DAE, do inglês *Differential Algebraic Equation*). A ideia é a de escolher um espaço de dimensão finita de soluções candidatas e um número de pontos no domínio (chamados *collocation points*), e selecionar uma solução que satisfaça a equação dada nos *collocation points*.

O *collocation method* é um caso particular do método Implícito de Runge-Kutta (IRK) onde o período de integração numérica $[0, T]$ (ou $[t_0, t_f]$) é particionado em n_e intervalos. O elemento finito i está definido no período $[t_{i-1}, t_i]$ e cada intervalo é aproximado por um polinômio de ordem n_c , onde n_c é o número de *collocation points*, sendo h_i o comprimento do intervalo. A curva é composta por n_e elementos finitos. O elemento finito pode ser observado na Figura 2.2.

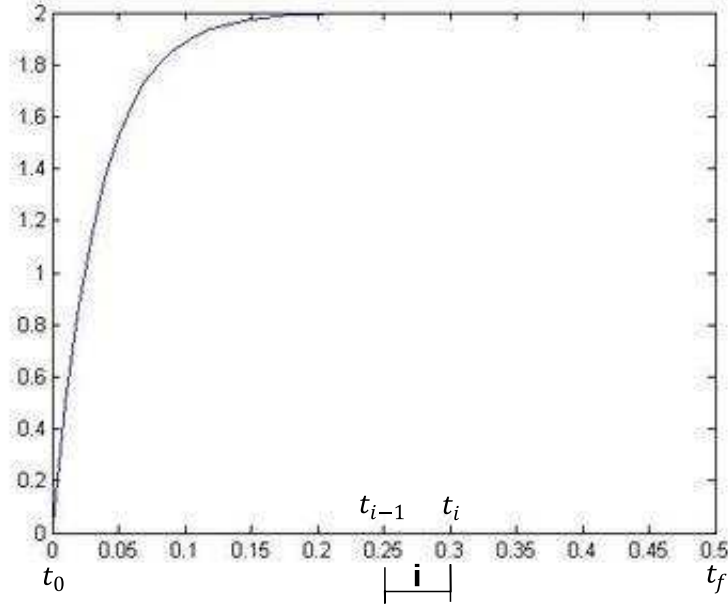


Figure 2.2: Elemento Finito i

Este polinômio, $\zeta_i(t)$ é a aproximação do elemento finito i que está definido no período $[t_{i-1}, t_i]$, pode ser representado de várias formas, ou seja, séries de potência, polinômio de Newton ou B-splines, no entanto, para desenvolver o método de colocação é preferível polinômios de interpolação de Lagrange. A razão é que esta classe de polinômios têm propriedades, que serão evidenciadas na próxima seção, de estabilidade, erro nulo para certos problemas e sua formulação é mais simples, pois os parâmetros do polinômio correspondem aos estados (*i.e.*; os parâmetros ζ_k da Eq. (2.8)) têm os mesmos limites que os estados, o que significa que uma restrição de estado pode ser aplicada diretamente sobre os coeficientes de aproximação. A Figura 2.3 demonstra um elemento finito e seus *collocation points*, apresentando também o valor de ζ_k para cada *collocation point*.

O polinômio base de Lagrange é dado por:

$$l_k(t) = \prod_{j=0, \neq k}^{n_c} \frac{(t - t_j)}{(t_k - t_j)} \quad (2.7)$$

t é a variável que seleciona um tempo específico no intervalo, sendo 0 o início do intervalo e 1 o fim, $t_k; k = 1, \dots, n_c$ o conjunto de *collocation points*.

A interpolação polinomial de Lagrange tem a forma geral:

$$\zeta(t) = \sum_{k=0}^{n_c} \zeta_k l_k(t) \quad (2.8)$$

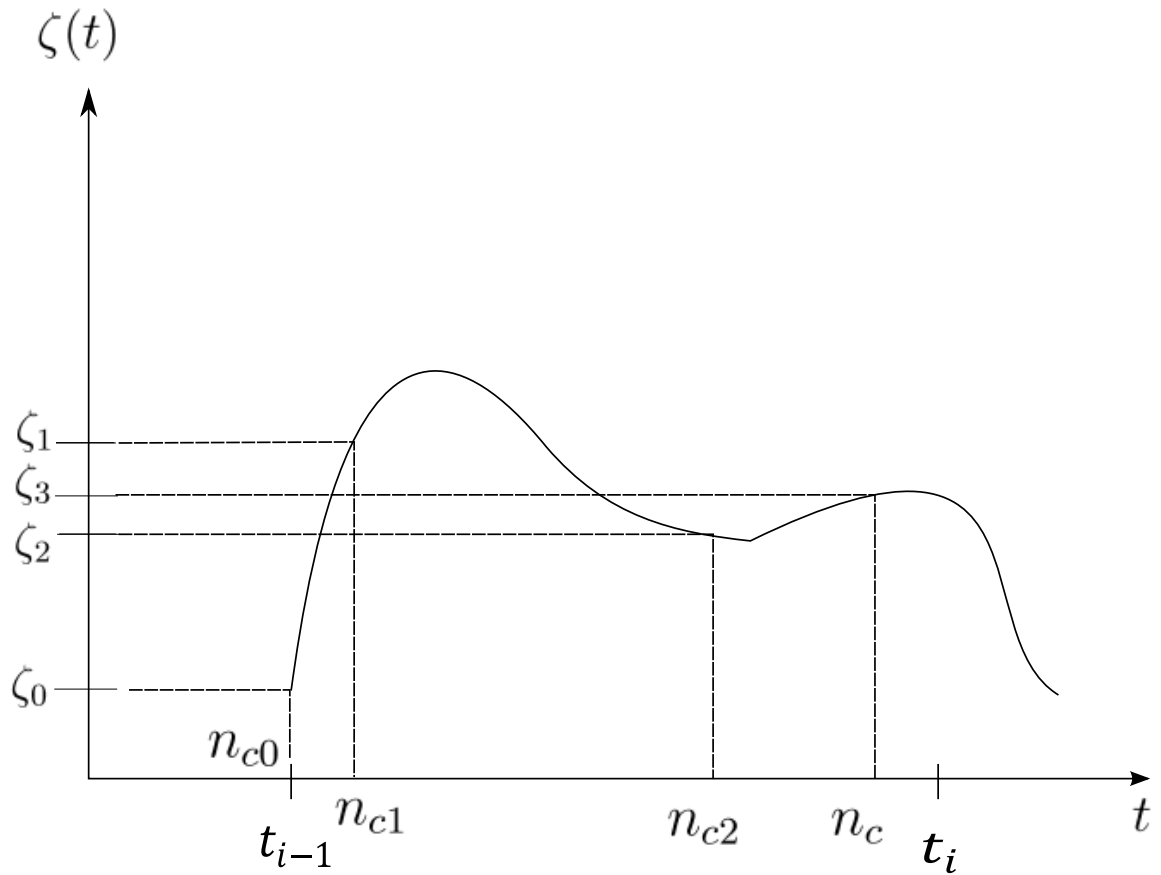


Figure 2.3: *Collocation points* em um elemento finito i

onde ℓ_k é o k -ésimo termo da base do polinômio de Lagrange, ζ_k é o coeficiente associado ao k -ésimo termo e n_c é o número de pontos de interpolação. Verificando a Equação 2.7, note que $\ell_k(t_k) = 1$, porém $\ell_k(t_j) = 0$, o que garante que a aproximação seja igual à curva nos *collocation points*.

A normalização do tempo permite utilizar os mesmos pontos de interpolação e polinômios base em todos os elementos de todas as variáveis, simplificando ainda mais o problema. A normalização é realizada como segue

$$t(\tau) = t_{i-1} + h_i \tau \quad (2.9)$$

onde t_i é o instante final do elemento i , chamado de ponto do malha de elemento i , e h_i é o comprimento do elemento i .

Derivando uma interpolação polinomial de Lagrange da Eq. (2.8) e utilizando a normalização do tempo da Eq. (2.9) a partir de agora, encontra-se:

$$\dot{\zeta}(\tau) = \sum_{k=0}^{n_c} \zeta_k \dot{\ell}_k(\tau) \quad (2.10)$$

e a derivada de ℓ_k é obtida através da regra do produto como:

$$\dot{\ell}_k(\tau) = \sum_{m=1, \neq k}^{n_c} \frac{1}{(\tau_k - \tau_m)} \prod_{j=0, \neq k, m}^{n_c} \frac{(\tau - \tau_j)}{(\tau_k - \tau_j)} \quad (2.11)$$

No elemento i o estado $x(t)$ é aproximado por $x_i(\tau)$, assim a aproximação só será contínua se

$$x_{i+1}(0) = x_i(1) \quad (2.12)$$

Uma vez que é necessário um ponto de interpolação, no início de cada elemento para x_i , a fim de garantir a continuidade, define-se

$$\tau_0 := 0 \quad (2.13)$$

sendo τ_0 o primeiro *collocation point* da interpolação.

Para a aproximação de \dot{x} no elemento i usa-se a derivada de x_i , com o tempo normalizado, os polinômios de colocação que representam as derivadas do estado são dados pela regra da cadeia, para isso é necessário diferenciar a Eq. (2.9). Como se segue:

$$dt = h_i d\tau \quad (2.14a)$$

$$\dot{x}_i(\tau) = \frac{d\tau}{dt} \frac{dx_i}{d\tau}(\tau) = \frac{1}{h_i} \sum_{k=0}^{n_c} x_{i,k} \dot{\ell}_k(\tau) \quad (2.14b)$$

onde $x_{i,k}$ é o k -ésimo *collocation point* do elemento finito i .

Seja

$$t_{i,k} := t_{i-1} + h_i \tau_k \quad (2.15a)$$

$$t_{i,k} : i \in [1 \dots n_e], k \in [1 \dots n_c] \quad (2.15b)$$

Então os *collocation points* coincidem com os pontos de interpolação dos polinômios de colocação, com a exceção de τ_0 em cada elemento. Esses pontos de interpolação são escolhidos como *collocation points*, porque pode-se facilmente obter todos os valores de x nesses pontos.

Assim a aproximação de x fica

$$x(t) = \sum_{k=0}^{n_c} \ell_k(\tau) x_{ik} \quad (2.16)$$

e sua derivada é

$$\frac{dx(t)}{d\tau} = \sum_{k=0}^{n_c} \frac{d\ell_k(\tau)}{d\tau} x_{ik} \quad (2.17a)$$

$$h_i f(x(t_{i,k}), t_{i,k}) = \sum_{k=0}^{n_c} \frac{d\ell_k(\tau)}{d\tau} x_{ik}, \quad k = 1, \dots, n_c \quad (2.17b)$$

No caso de $n_c > 1$, é preciso garantir a continuidade do sistema

$$x_{1,0} = x_0 \quad (2.18a)$$

$$x_{i+1,0} = \sum_{k=0}^{n_c} \ell_k(1) x_{ik}, \quad \forall i \in [1, n_e] \quad (2.18b)$$

$$x_f = \sum_{k=0}^{n_c} \ell_k(1) x_{n_e k} \quad (2.18c)$$

onde $x_{1,0}$ representa o valor do primeiro estado, no início do primeiro sub-intervalo, $x_{i+1,0}$ é o valor de estado, no início do intervalo i e x_f é o valor de estado, no final do período de simulação.

Generalizando o problema para a interpolação não somente dos estados, constrói-se um vetor z , definido por

$$z(t) = (x(t), u(t), w(t)) \quad (2.19)$$

onde $x(t)$ são os estados, $u(t)$ o sinal de controle e $w(t)$ variáveis algébricas. No elemento i são aproximados por um vetor de interpolação polinomial de Lagrange

$$z_i(\tau) = (x_i(\tau), u_i(\tau), w_i(\tau)) \quad (2.20)$$

Assim, as equações para o vetor $z(t)$ são análogas às dos estados $x(t)$.

2.4: Polinômio Deslocado de Gauss-Jacobi

A escolha dos pontos de interpolação restantes é o que define o *collocation method* específico. Essa escolha é baseada nas raízes do polinômio deslocado de Gauss-Jacobi de grau $K = n_c - \alpha - \beta$, que pode ser descrito por

$$P_K^{(\alpha, \beta)}(\tau) = \sum_{j=0}^K (-1)^{K-j} \gamma_j \tau^j \quad (2.21)$$

onde,

$$\gamma_0 = 1 \quad (2.22a)$$

$$\gamma_j = \frac{(K+1-j)(K+j+\alpha+\beta)}{j(j+\beta)}, j \in [1, \dots, K] \quad (2.22b)$$

onde α e β são constantes a escolher, definidas no domínio $[0, 1]$. O polinômio é deslocado no sentido que normalmente está definido com $\tau \in [-1, 1]$, enquanto aqui ele está definido no domínio $[0, 1]$. A combinação das constantes forma 3 métodos diferentes.

$$\alpha = 0 \text{ e } \beta = 0 \rightarrow \text{Gauss collocation} \quad (2.23a)$$

$$\alpha = 1 \text{ e } \beta = 0 \rightarrow \text{Radau collocation} \quad (2.23b)$$

$$\alpha = 1 \text{ e } \beta = 1 \rightarrow \text{Lobatto collocation} \quad (2.23c)$$

O método Lobatto, diferente do Radau e do Gauss, possui um *collocation point* no início de cada elemento, ou seja, τ_0 é obtido através da Eq. (2.21) e não pode ser definido como feito na Eq (2.13). Por esta razão, o método descrito nesta seção não pode ser usado para criar um método Lobatto. É necessária uma abordagem diferente para o Lobatto *collocation*.

Como exemplo a solução da Eq. (2.21) é dada para n_c de 1 a 4, na Tabela 2.1 para o Radau e na Tabela 2.2 para o Gauss. As soluções apresentadas estão arredondadas para melhor comparar a diferença entre os dois métodos. Para n_c maiores pode-se utilizar de computadores para a resolução da Eq. (2.21)

Table 2.1: Radau Collocation Points ($\alpha = 1$ e $\beta = 0$)

	τ_1	τ_2	τ_3	τ_4
$n_c = 1$	1,000	-	-	-
$n_c = 2$	0,333	1,000	-	-
$n_c = 3$	0,155	0,644	1,000	-
$n_c = 4$	0,088	0,409	0,787	1,000

Table 2.2: Gauss Collocation Points ($\alpha = 0$ e $\beta = 0$)

	τ_1	τ_2	τ_3	τ_4
$n_c = 1$	0,500	-	-	-
$n_c = 2$	0,211	0,788	-	-
$n_c = 3$	0,112	0,500	0,887	-
$n_c = 4$	0,069	0,330	0,669	0,930

2.5: Ordem de Convergência

Como mostrado em [3] e [10] o erro global uniforme do estado aproximado no elemento i , é

$$O(h_i^{\min(n_c+1, 2n_c-\alpha-\beta)}) \quad (2.24)$$

O erro global uniforme das variáveis algébricas aproximadas e variáveis de controle no elemento i é

$$O(h_i^{n_c}) \quad (2.25)$$

Se $n_c > 1$ o erro global uniforme é, portanto, o mesmo para Gauss, Radau e Lobatto.

No final/começo dos elementos, o erro global de todas as variáveis é

$$O(h_i^{2n_c-\alpha-\beta}) \quad (2.26)$$

Assim, a ordem de convergência da solução é de até duas vezes mais elevada no final/começo dos elementos do que no resto do elemento (incluindo os pontos de colocação). Este fenômeno é chamado super convergência.

2.6: Exemplo

Define-se o sistema:

$$\frac{dx}{dt} = x^2 - 2x + 1, \quad x(0) = -3 \quad (2.27)$$

com $t \in [0, 1]$. A solução analítica desse sistema é dada pela equação $x(t) = (4t - 3)/(4t + 1)$. Entretanto serão calculadas aproximações numéricas usando o *collocation method* utilizando as raízes do polinômio (2.21) de Radau e Gauss com $n_c = 3$.

2.6.1: Radau Collocation

Os *collocation points* são tirados do polinômio da Eq. (2.21), substituindo os valores de α e β do Radau collocation na Eq. (2.22) e esta na Eq. (2.21), obtém-se

$$P_K^{Radau}(\tau) = 1 + \sum_{j=1}^2 (-1)^{2-j} \frac{(3-j)(3+j)}{j^2} \tau^j \quad (2.28)$$

Com Radau os *collocation points* são $\tau_0 = 0$, $\tau_1 = 0.155051$, $\tau_2 = 0.644949$, e $\tau_3 = 1$. A Figura 2.4 demonstra os *collocation points* em relação com a resposta analítica

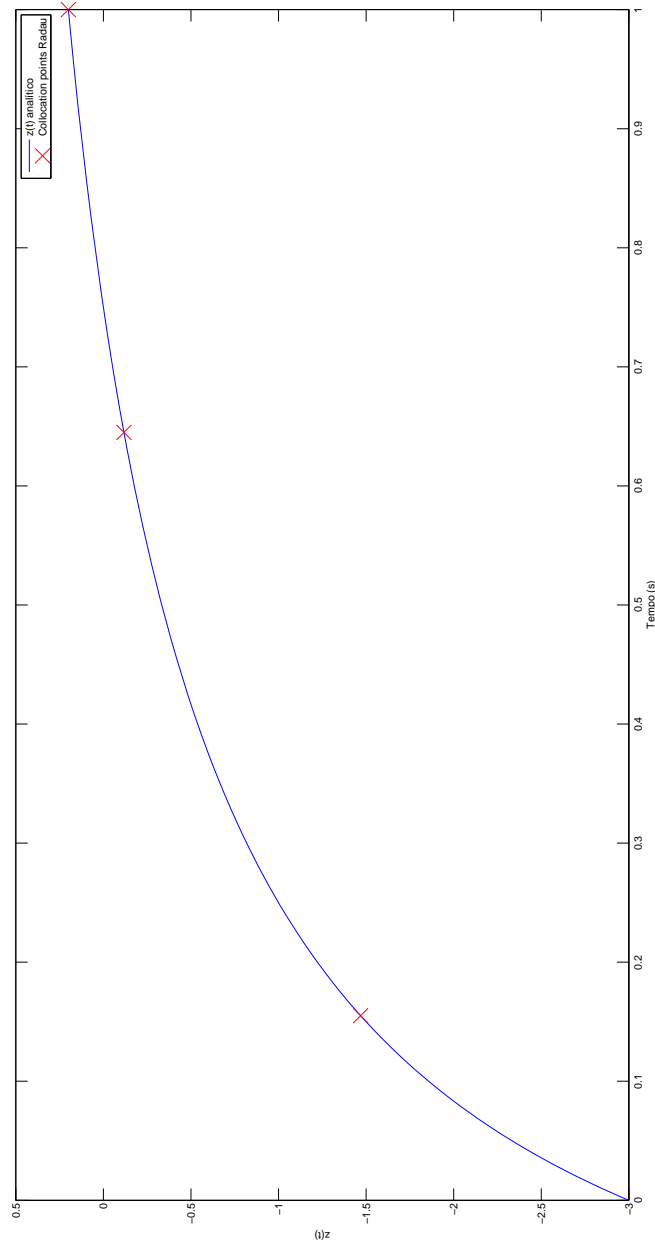


Figure 2.4: *Collocation points* do método de Radau

Usando a Eq. (2.17b) com $N = 1$ elementos, e $h_i = 1/N$ obtém-se

$$\sum_{j=0}^3 z_{ij} \frac{d\ell_j(\tau_k)}{d\tau} = h_i(z_{ij}^2 - 2z_{ik} + 1), \quad k = 1, \dots, 3, \quad i = 1 \quad (2.29)$$

Desenvolvendo a base de Lagrange descrita na Eq. (2.11), encontra-se:

$$\frac{d\ell_0(\tau_k)}{d\tau} = -30\tau_k^2 + 36\tau_k - 9 \quad (2.30a)$$

$$\frac{d\ell_1(\tau_k)}{d\tau} = 46.7423\tau_k^2 - 51.2392\tau_k - 10.0488 \quad (2.30b)$$

$$\frac{d\ell_2(\tau_k)}{d\tau} = -23.7423\tau_k^2 + 20.5925\tau_k - 1.38214 \quad (2.30c)$$

$$\frac{d\ell_3(\tau_k)}{d\tau} = 10\tau_k^2 - \frac{16}{3}\tau_k + \frac{1}{3} \quad (2.30d)$$

Substituindo a Eq. (2.30) na Eq. (2.29) encontra-se:

$$\begin{aligned} & z_0(-30\tau_k^2 + 36\tau_k - 9) + z_1(46.7423\tau_k^2 - 51.2392\tau_k + 10.0488) \\ & z_2(-23.7423\tau_k^2 + 20.5925\tau_k - 1.38214) + z_3\left(10\tau_k^2 - \frac{16}{3}\tau_k + \frac{1}{3}\right) \\ & = (z_k^2 - 2z_k + 1), \quad k = 1, \dots, 3 \end{aligned} \quad (2.31)$$

Resolvendo o sistema de equações tem-se $z_1 = -1.65701$, $z_2 = 0.032053$, $z_3 = 0.207272$ com $z_0 = -3$.

Um gráfico na Figura 2.6 mostra a comparação do método de Radau com o de Gauss, que será descrito na próxima subseção, e com a solução analítica.

2.6.2: Gauss Collocation

Os *collocation points* são tirados do polinômio da Eq. (2.21), substituindo os valores de α e β do Gauss collocation na Eq. (2.22) e esta na Eq. (2.21), obtém-se

$$P_K^{Gauss}(\tau) = -1 + \sum_{j=1}^3 (-1)^{3-j} \frac{(4-j)(3+j)}{j^2} \tau^j \quad (2.32)$$

Com Gauss os *collocation points* são $\tau_0 = 0$, $\tau_1 = 0.112702$, $\tau_2 = 0.500000$, e $\tau_3 = 0.887298$. A Figura 2.5 demonstra os *collocation points* em relação com a resposta analítica

O equacionamento desenvolvido na Subseção 2.6.1 não se altera, por exceção do conjunto de equações em (2.31), pois os valores de τ_k são dados pela Eq. (2.32).

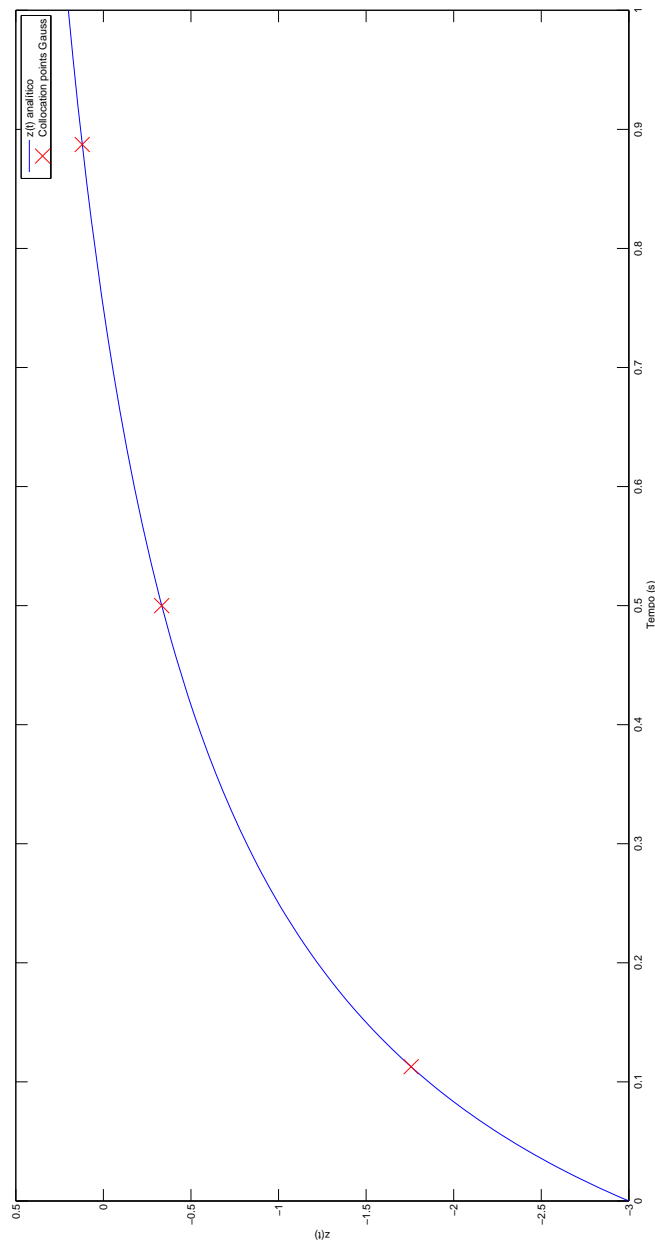


Figure 2.5: *Collocation points* do método de Gauss

Resolvendo o sistema de equações, tem-se $z_1 = -1.742931$, $z_2 = -0.2488663$, $z_3 = 0.176679$ com $z_0 = -3$.

A Figura 2.6 mostra os 3 métodos considerados: solução analítica, aproximação com *Gauss collocation* e aproximação com *Radau collocation*.

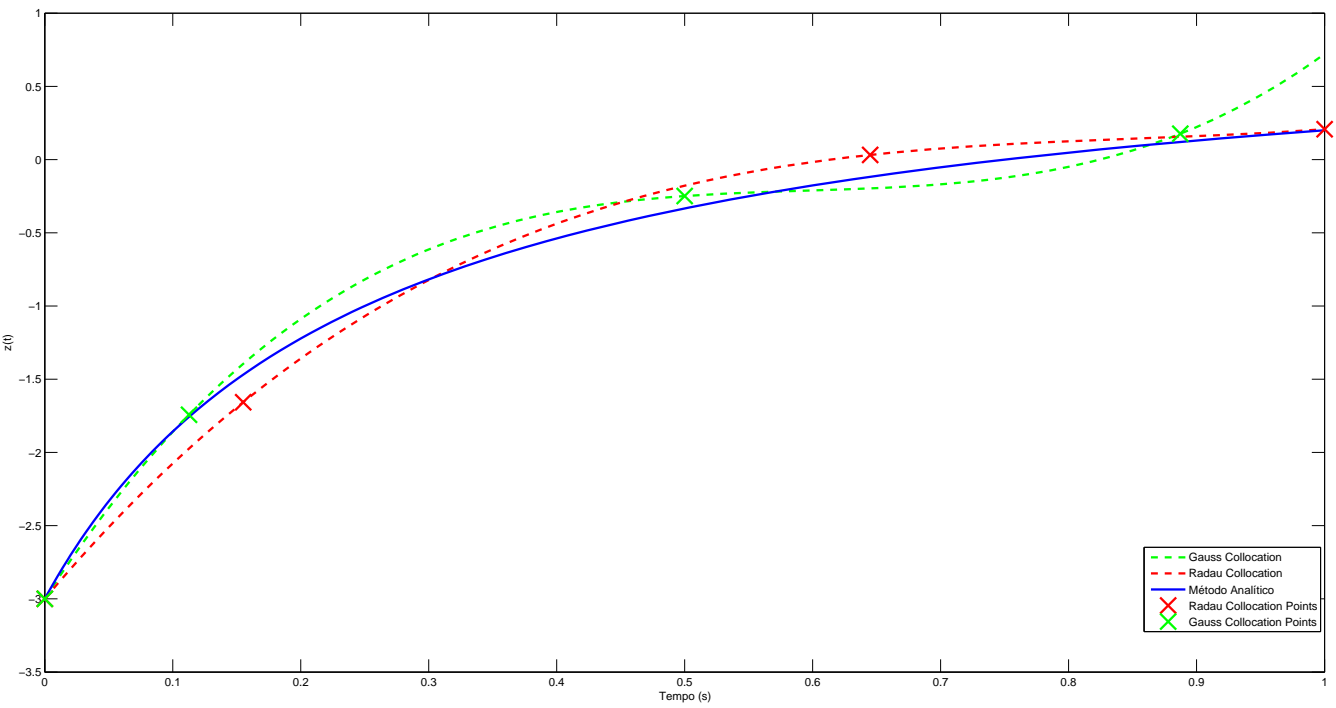


Figure 2.6: Gráfico das respostas

A Tabela 2.3 contem o erro de aproximação de Gauss e de Radau em compa-

ração com a solução analítica. O erro foi calculado com a seguinte formula:

$$\text{Erro}_{\text{gauss}} = \int_0^1 (z(t) - z_g(t))^2 dt \quad (2.33a)$$

$$\text{Erro}_{\text{radau}} = \int_0^1 (z(t) - z_r(t))^2 dt \quad (2.33b)$$

sendo $z_g(t)$ a aproximação da curva pelo Gauss *collocation*, $z_r(t)$ a aproximação da curva pelo Radau *collocation* e $z(t)$ a resposta analítica.

Table 2.3: Erro quadrático das curvas

Erro aproximação de Gauss	Erro aproximação de Radau
0,0220575614331819	0,0153388854412689

Chapter 3: Ferramentas de Implementação

Esse capítulo apresenta as ferramentas utilizadas para modelar e resolver o problema de estimação de parâmetros.

3.1: Modelica

Modelica é uma linguagem orientada a objetos desenvolvida para modelar o comportamento dinâmico dos sistemas técnicos de uma forma simples. Os modelos são descritos usando equações diferenciais, algébricas e discretas. Os modelos também podem ser compostos por outros modelos em uma estrutura hierárquica. Modelica consegue lidar com problemas grandes e complexos devido a sua capacidade de reutilização de modelos.

Modelica é uma linguagem textual, no entanto, existem vários ambientes que fornecem uma interface gráfica ao usuário. Algumas implementações são gratuitas e outras comerciais. Ambientes livres são: OpenModelica (Linköping University) e JModelica.org (Universidade de Lund). Ambientes comerciais são: CATIA Systems (Dassault Systemes), Dymola (Dynasim), MapleSim (MAPLESOFT), MathModelica (Wolfram Research), e outros.

3.2: Optimica

Enquanto o uso primário de modelos Modelica é simulação, vários outros usos estão surgindo. Como não é viável recodificar os modelos para cada novo modelo de uso, futuras ferramentas Modelica e também a própria linguagem Modelica, devem acomodar e promover novos usos de modelos Modelica.

Um exemplo de uso emergente de modelos Modelica é otimização dinâmica. Uma característica de problemas de otimização dinâmica realista é que o procedimento de formulação de tais problemas é altamente interativo. É comum que uma extensa afinação da função de custo e restrições seja necessária, a fim de obter uma solução aceitável. Esses elementos são indispensáveis na formulação de problemas de otimização dinâmica e não são suportados pela linguagem Modelica.

A fim de formular um problema de otimização dinâmica, a ser resolvido por um algoritmo numérico, o usuário tem de fornecer diferentes tipos de informação. É natural categorizar essas informações em três níveis, o que corresponde ao aumento dos níveis de detalhe [1].

- **Nível I.** No nível matemático, uma formulação canônica de um problema de otimização dinâmica é dado. Isto inclui variáveis e parâmetros para otimizar, função de custo para minimizar, restrições e modelo Modelica constituindo a restrição dinâmica. O problema formulado é, em geral, infinito no aspecto de, que não pode ser usado diretamente através de um algoritmo numérico, sem informação adicional.
- **Nível II.** No nível da transcrição, um método para traduzir o problema a partir de um problema de dimensão infinita para um problema de dimensão finita precisa ser fornecido. Isto pode incluir malhas de discretização, bem como estimativas iniciais para os parâmetros de otimização e variáveis. Deve ser notado que a informação necessária para este nível depende do algoritmo numérico que é utilizado para resolver o problema.
- **Nível III.** No nível do algoritmo, informações tais como tolerâncias e os parâmetros de controle do algoritmo podem ser fornecidas. Tais parâmetros são muitas vezes críticos, de modo a obter um desempenho aceitável em termos de convergência, confiabilidade numérica e velocidade.

Em Modelica, apenas as informações correspondentes ao Nível I são expressas na descrição real do modelo. Para otimização dinâmica, a necessidade de entrada do usuário no nível de algoritmo é mais enfatizada. Algoritmos automáticos, por exemplo para a seleção de malha, existem, mas podem não serem adequados para todos os tipos de problemas. Por conseguinte, é desejável incluir, na linguagem, meios para o usuário especificar a maior parte dos aspectos do problema de modo a manter a flexibilidade, permitindo ao mesmo tempo o uso de algoritmos automáticos quando possível e adequado.

A extensão Optimica permite ao usuário especificar elementos importantes de um problema de otimização dinâmica, tais como funções de custo, restrições e intervalo de otimização. O modelo dinâmico, no qual o problema de otimização dinâmica é baseado, é expresso usando o padrão Modelica. Optimica também suporta uma anotação que permite ao utilizador especificar as propriedades de um método de

transcrição, baseado na colocação direta. Devido a estas propriedades, Optimica suporta formulação de problemas de otimização dinâmica, utilizando construções de alto nível, tanto no nível matemático quanto no nível da transcrição numérica.

Optimica apresenta três aspectos principais no padrão Modelica:

- Classe de otimização, uma nova classe de objetos pode ser instanciada. Ela representa o problema de otimização. Necessariamente precisa de uma função objetivo e variáveis de otimização. É importante notar que o horizonte de predição pode ser uma variável de otimização.
- Uma seção de restrição dentro de elementos de otimização. Nesta seção, podem-se declarar igualdades e desigualdades que impõem restrições às variáveis.
- Variáveis de acesso em tempo. Com o padrão Modelica não podemos acessar as variáveis em um momento específico. Com Optimica é possível fazer restrições como: $x(t_f) = 1$ (variável x no tempo final t_f deve ser igual a 1) ou $y(5) \leq 3$ (variável y no tempo de 5 deve ser menor do que 3).

3.3: JModelica.org

JModelica.org é um pacote para simulação e otimização de modelos Modelica. É desenvolvido em colaboração entre a indústria e acadêmicos, com a finalidade de criar uma plataforma industrialmente viável usando o estado da arte dos algoritmos para analisar sistemas físicos complexos. Python é usado como uma linguagem de programação para criar uma interface amigável para todos os componentes do JModelica.org [12].

O ambiente JModelica consiste em um conjunto de módulos de software, incluindo compiladores para Modelica e Optimica, um gerador de código para linguagem C, uma biblioteca de tempo de execução em C, um algoritmo de otimização simultânea, e uma biblioteca para integração com a linguagem de programação Python. Também, internamente, ele se conecta com outras três ferramentas muito usadas: CasADi, Sundials, e IPOPT [15].

É típico que a solução de problemas de otimização dinâmicos exigem múltiplas iterações, em que a função de custo, as restrições, o método da transcrição, e mesmo o modelo são refinados de modo a obter melhores resultados. Os resultados, então,

normalmente precisam ser analisados e os parâmetros de formulação de otimização ajustados. O uso de linguagens de descrição de alto nível libera o usuário da tarefa custosa e propensa a erros de codificação das formulações de modelos e otimização em linguagens menos adequadas. Com efeito, o foco do processo de concepção é transferido da codificação do problema para a formulação do problema, que se traduz em processos de projeto mais eficientes.

3.3.1: CasADi

CasADi é um sistema minimalista de álgebra computacional que implementa diferenciação automática, por meio de uma abordagem híbrida simbólica/numérica. Ele é projetado para ser uma ferramenta de baixo nível para a implementação rápida, mas altamente eficiente, de algoritmos de otimização numérica.

Em otimização é importante calcular eficientemente derivados de função. Para isso usa-se CasADi (Computer algebra system with Automatic Differentiation). Uma vez que uma representação simbólica que consiste em objetos CasADi de um problema de NLP foi criada, CasADi fornece todas as informações das derivadas necessárias para a solução numérica do problema com muito pouco esforço do usuário.

3.3.2: Sundials

Sundials é um pacote de *solvers* de ODE e DAE que foi desenvolvido para ser um integrador de tempo robusto e *solver* não-linear. O *solver* mais utilizado é CVODE, um *solver* de ODE com a capacidade de resolver alguns tipos especiais de DAE que podem calcular automaticamente sensibilidades.

3.3.3: IPOPT

Ipop (Interior Point OPTimizer) é um pacote de software para otimização não-linear em grande escala. Ele é projetado para encontrar soluções (locais) de problemas de otimização matemática.

Ipop usa um algoritmo de pontos interiores com um filtro do método de *line-search*. Ele também tem uma função automática de problema de escala, algumas heurísticas para acelerar a convergência, e algumas técnicas para aumentar a robustez.

A Figura 3.1 mostra a esquematização da estrutura dos softwares utilizados na resolução de problemas de otimização.

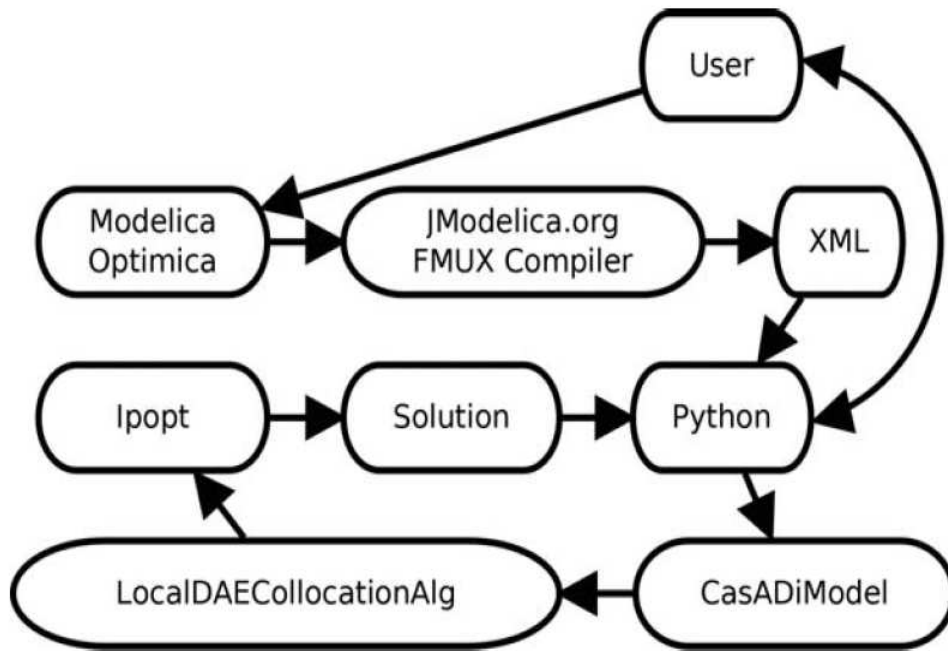


Figure 3.1: Visão geral da estrutura de otimização

Chapter 4: Estimação de Parâmetros

Neste Capítulo apresenta-se a definição de técnica de otimização e se aplica o *collocation method*, descrito no Capítulo 2, juntamente com essa nova técnica em um circuito RLC para a validação da metodologia.

Assim que validada, muda-se o sistema: a metodologia é aplicada em um poço produtor de petróleo e gás. Neste caso, foi possível obter resultados positivos em relação aos parâmetros que serão descritos neste Capítulo. Os resultados impulsionaram a utilização de dados de simuladores mais sofisticados para serem usados como referência da metodologia, porém não se obtiveram bons resultados, por isso foi necessária a modificação do modelo, tornando-o levemente mais sofisticado.

Todos os códigos utilizados para os modelos e para a resolução dos problemas de estimação se encontram no apêndice B.

4.1: Otimização

No caso mais simples, um problema de otimização consiste em maximizar ou minimizar uma função real, escolhendo sistematicamente valores de entrada a partir de um conjunto permitido e calculando o valor da função. A generalização da teoria de otimização e técnicas para outras formulações compreende uma grande área da matemática aplicada. De modo mais geral, a otimização busca encontrar “o melhor valor possível” de alguma função objetivo dado um domínio definido (ou um conjunto de restrições), incluindo uma variedade de diferentes tipos de funções objetivo e diferentes tipos de domínios. Problemas de otimização podem ser divididos em duas categorias, dependendo se as variáveis são contínuas ou discretas. Um problema de otimização com variáveis discretas é conhecido como um problema de otimização combinatória.

Um problema de otimização dinâmica normalmente tem a seguinte forma:

$$\min \psi(t, x(t), y(t), w(t), u(t)) \quad (4.1a)$$

$$\text{s.t.: } \dot{x} = f(t, x(t), w(t), u(t), \theta) \quad (4.1b)$$

$$g(t, x(t), y(t), u(t)) = 0 \quad (4.1c)$$

$$x^L \leq x(t) \leq x^U \quad (4.1d)$$

$$y^L \leq y(t) \leq y^U \quad (4.1e)$$

$$w^L \leq w(t) \leq w^U \quad (4.1f)$$

$$u^L \leq u(t) \leq u^U \quad (4.1g)$$

$$\theta^L \leq \theta(t) \leq \theta^U \quad (4.1h)$$

$$t_0 \leq t \leq t_f \quad (4.1i)$$

onde

- ψ é a função objetivo.
- $x(t)$, $y(t)$, $w(t)$, $u(t)$ e θ são os estados, saída, variáveis algébricas e de controle e os parâmetros da função, respectivamente.
- f e g são funções de estado e funções algébricas, respectivamente.
- x^L e x^U são os limites inferiores e superiores, respectivamente, dos estados.
- y^L e y^U são os limites inferiores e superiores, respectivamente, da saída.
- w^L e w^U os limites inferiores e superiores, respectivamente, das variáveis algébricas.
- u^L e u^U são os limites inferiores e superiores, respectivamente, do controle.
- θ^L e θ^U são os limites inferiores e superiores, respectivamente, dos parâmetros.

Essas variáveis podem ser discretizadas utilizando o *collocation method* mostrado nas seções anteriores. Os limites de variáveis são os mesmos da formulação de tempo contínua, os quais são aplicados diretamente sobre os *collocation points*. Esta facilidade de aplicação de restrições é uma das razões pelas quais o polinômio de interpolação de Lagrange é escolhido. No entanto, essas restrições só podem ser asseguradas nos *collocation points* e não no intervalo entre eles.

4.1.1: Função Objetivo

A formulação mais comum e flexível para a função objetivo integra o erro quadrático ponderado entre os estados, as variáveis algébricas, os controles e suas referências. Esta formulação permite o controle de diferentes variáveis do sistema.

$$\min \psi = \int_{t_0}^{t_f} [(x - x_{ref})^T Q_x (x - x_{ref}) + (y - y_{ref})^T Q_y (y - y_{ref}) + (w - w_{ref})^T Q_w (w - w_{ref}) + (u - u_{ref})^T Q_u (u - u_{ref})] dt \quad (4.2)$$

onde Q_x , Q_y , Q_w e Q_u são os fatores de peso dos estados, saídas, variáveis algébricas e controle, respectivamente.

No caso do problema de estimação de parâmetros, as referências são obtidas pelos dados medidos do sistema. Os dados são compostos das entradas e saídas do sistema. Para resolver o problema adicionam-se restrições às entradas e aos estados, entretanto o mesmo não pode ser feito com as saídas. Essas restrições de igualdade são aplicadas apenas nos *collocation points*, assim forçando as entradas a serem iguais aos dados de referência. Nesse caso, os parâmetros possuem um valor livre, para que o algoritmo possa encontrar o valor que minimize a função objetivo e cumpra com as restrições. Adicionando as novas restrições e discretizando com o *collocation method* as Eqs. (4.1), obtém-se:

$$\min_{\theta} \psi = \sum_{i=0}^{n_e} \sum_{k=0}^{n_c} [(x_{ik} - x_{ref}(t_{ik}))^T Q_x (x_{ik} - x_{ref}(t_{ik})) + (y_{ik} - y_{ref}(t_{ik}))^T Q_y (y_{ik} - y_{ref}(t_{ik})) + (w_{ik} - w_{ref}(t_{ik}))^T Q_w (w_{ik} - w_{ref}(t_{ik})) + (u_{ik} - u_{ref}(t_{ik}))^T Q_u (u_{ik} - u_{ref}(t_{ik}))]$$
(4.3a)

$$\text{s.t.: } \sum_{k=0}^{n_c} \frac{d\ell_k(\tau)}{d\tau} x_{ik} - h_i f(t_{ik}, x_{ik}, u_{ik}, \theta) = 0, \forall k \in [1, n_c], i \in [1, n_e]$$
(4.3b)

$$g(t_{ik}, x_{ik}, u_{ik}) = 0, \forall k \in [1, \dots, n_c], i \in [1, \dots, n_e]$$
(4.3c)

$$x_{i+1,0} = \sum_{k=0}^{n_c} \ell_k(1) x_{ik}, \forall i \in [1, \dots, n_e]$$
(4.3d)

$$x_f = \sum_{k=0}^{n_c} \ell_k(1) x_{n_e k}$$
(4.3e)

$$x^L \leq x_{ik} \leq x^U$$
(4.3f)

$$y^L \leq y_{ik} \leq y^U$$
(4.3g)

$$w^L \leq w_{ik} \leq w^U$$
(4.3h)

$$u^L \leq u_{ik} \leq u^U$$
(4.3i)

$$u_{ik} = u_{ref}(t_{ik})$$
(4.3j)

$$x_{ik} = x_{ref}(t_{ik})$$
(4.3k)

$$x_{1,0} = x_0$$
(4.3l)

em que τ encontrado dependerá do tipo do *collocation method*, utilizando as Tabelas 2.1 e 2.2. A Figura 4.1 demonstra um exemplo, apontando onde se encontram os *collocation points*, elementos finitos e o período da aproximação.

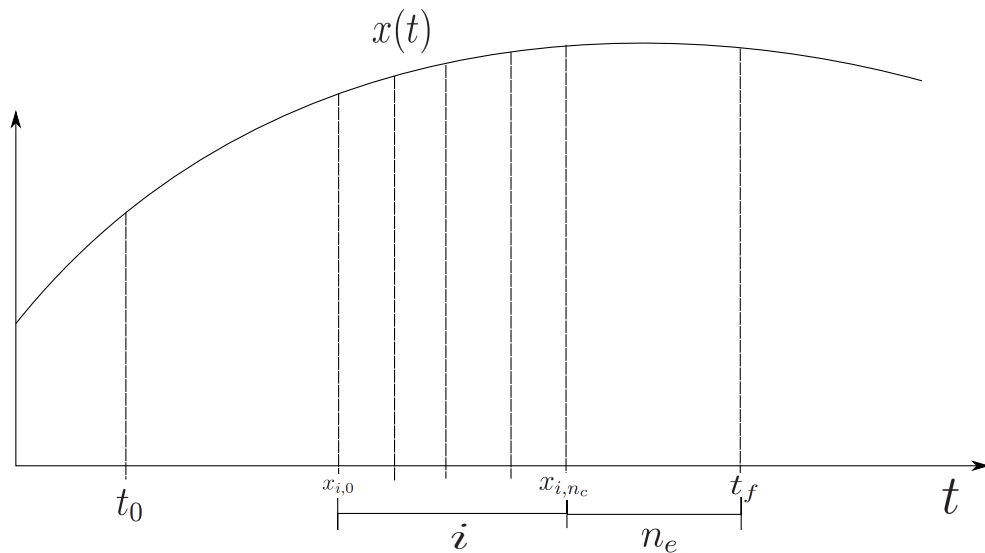


Figure 4.1: *Collocation points*, elementos finitos e período de aproximação

4.2: Circuito RLC

Um circuito RLC é um circuito elétrico constituído por uma resistência, um indutor e um capacitor, ligado em série ou em paralelo. O circuito forma um oscilador harmônico para a corrente que ressoa de forma semelhante a um circuito LC. A principal diferença da presença da resistência é que ela faz com que qualquer oscilação induzida no circuito irá extinguir-se ao longo do tempo se não for mantida por uma fonte. Este efeito da resistência é chamado de amortecimento. A presença da resistência também reduz ligeiramente a frequência ressonante de pico. Alguma resistência é inevitável em circuitos reais, mesmo se uma resistência não está especificamente incluída como um componente.

Há muitas aplicações para este circuito. Eles são utilizados em diversos tipos diferentes de circuitos osciladores. Outra aplicação importante é para *tuning*, como em receptores de rádio ou aparelhos de televisão, onde são usados para selecionar uma estreita faixa de frequências de ondas de rádio do ambiente. Um circuito RLC pode ser usado como um filtro passa-banda, filtro rejeita-faixa, filtro passa-baixa ou filtro passa-alta. A Figura 4.2 apresenta um modelo do circuito RLC em série.

A equação diferencial que regula o circuito pode ser encontrada através da lei

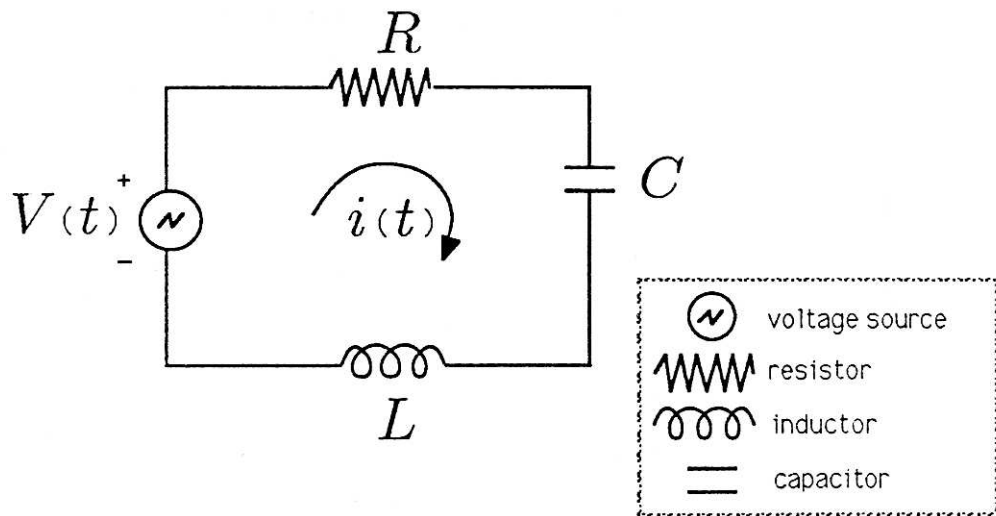


Figure 4.2: Circuito RLC.

de Kirchhoff.

$$V_R + V_L + V_C = V(t) \quad (4.4a)$$

$$Ri + L \frac{di}{dt} + V_C = V(t) \quad (4.4b)$$

Pela equação do capacitor temos:

$$\frac{dV_c}{dt} = \frac{i}{C} \quad (4.5)$$

Com isso podemos montar um sistema de equações de estado

$$x_1 = V_c(t) \quad (4.6a)$$

$$x_2 = i(t) \quad (4.6b)$$

Substituindo as equações (4.6) nas equações (4.4b) e (4.5) encontra-se o sis-

tema abaixo

$$\dot{x}_1 = \frac{x_2}{C} \quad (4.7a)$$

$$\dot{x}_2 = \frac{V(t)}{L} - \frac{x_1}{L} - \frac{Rx_2}{L} \quad (4.7b)$$

4.2.1: Estimação dos Parâmetros do Circuito RLC

Utilizando o modelo descrito anteriormente, propõe-se um problema de estimação dos parâmetros do circuito, R , L e C utilizando o *collocation method* e técnicas de otimização. Para isso primeiramente são necessários dados que são obtidos a partir de simulações. O circuito usado na simulação tem os parâmetros: $R = 200\Omega$, $L = 0,01H$, $C = 1e^{-3}F$ e uma entrada $V(t) = 5V$.

Uma vez com os dados e o conhecimento do modelo, simula-se o modelo com uma estimativa inicial dos parâmetros. Essa simulação fornecerá a trajetória de otimização usada na estimação. Após isso, utiliza-se o *collocation method* nos dados amostrados de $V(t)$, $Vc(t)$ e $I(t)$, reconstruindo as curvas para então poder estimar os valores de R , L e C .

Abaixo encontra-se a Tabela 4.1 com as diferentes situações de simulação, com uma estimativa inicial de $R = 400\Omega$, $L = 0,02H$ e $C = 2e^{-3}F$.

Table 4.1: Parâmetros Estimados

El fin	CP	T (s)	R (Ω)	L (H)	C (F)	Erro R (%)	Erro L (%)	Erro C (%)
			200,00	0,01	0,001			
1000	1	54,59	199,90	0,00944	0,0009756	0,047	5,590	2,432
1000	10	465,67	200,36	0,00966	0,0009981	0,183	3,345	0,182
100	1	1,18	199,88	0,01543	0,0007711	0,055	54,360	22,881
100	10	7,55	201,70	0,01938	0,0009916	0,850	93,895	0,832

Pode-se perceber que existe uma troca entre o número de elementos finitos (El fin) e o número de *collocation points*(CP), uma vez que mais elementos finitos aumenta o tempo de processamento, mas também aumenta a precisão da estimação. O mesmo ocorre com o número de *collocation points*. Entretanto, uma escolha ruim na razão entre esses parâmetros pode ocasionar um aumento desnecessário de tempo de processamento e uma pior estimação.

Para melhor estudar esse método de estimação de parâmetros, altera-se a entrada, até então constante para uma série de degraus, conforme ilustra a Figura 4.3.

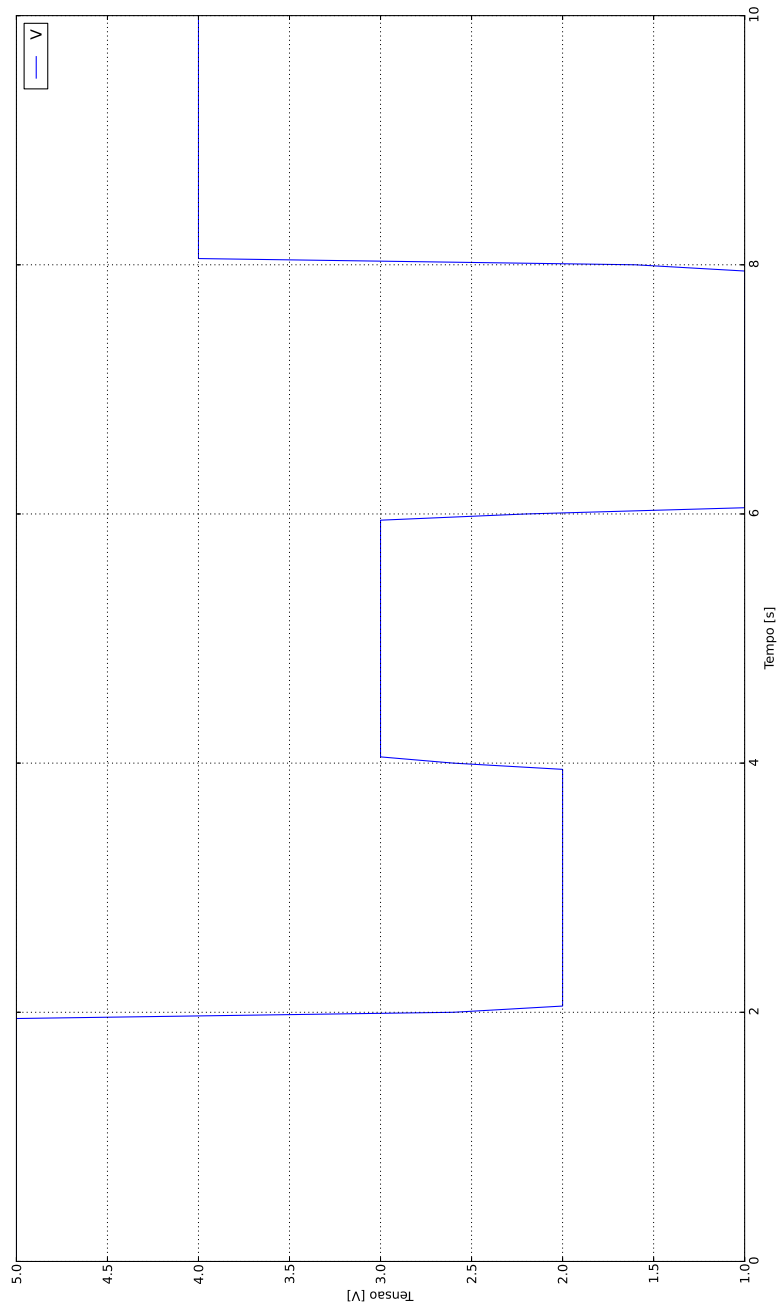


Figure 4.3: Entrada $V(t)$.

Assim, aplicando a mesma metodologia para o caso de entrada constante, com

os mesmos valores de parâmetros e estimativas iniciais encontram-se os resultados mostrados na Tabela 4.2.

Table 4.2: Parâmetros Estimados

El fin	CP	T (s)	R (Ω)	L (H)	C (F)	Erro R (%)	Erro L (%)	Erro C (%)
			200,00	0,01	0,001			
100	1	1,04	196,62	0,0004016	0,00089185	1,69	95,984	10,815
200	1	1,78	198,26	0,0005798	0,00094373	0,87	94,202	5,627
1000	1	28,85	199,50	0,0067239	0,00098881	0,25	32,761	1,119

Apesar de haver um grande erro na estimação de L , comparando as respostas do sistema com os dois conjuntos de parâmetros nas Figuras 4.4, 4.5, 4.6 e 4.7, observa-se que elas são equivalentes. Na Tabela 4.3 apresenta-se o erro de aproximação do resultado encontrado pelo *collocation method* em comparação com o sistema simulado. O erro foi calculado com a seguinte formula:

$$\text{Erro}_I = \int_0^{t_f} (I(t) - I^p(t))^2 dt \quad (4.8a)$$

$$\text{Erro}_{V_c} = \int_0^{t_f} (V_c(t) - V_c^p(t))^2 dt \quad (4.8b)$$

$$\text{Erro}_{V_r} = \int_0^{t_f} (V_r(t) - V_r^p(t))^2 dt \quad (4.8c)$$

sendo t_f o tempo final de simulação, $I(t)$, $V_c(t)$ e $V_r(t)$ a corrente, tensão no capacitor e tensão no resistor do sistema simulado e $I^p(t)$, $V_c^p(t)$ e $V_r^p(t)$ a corrente, tensão no capacitor e tensão no resistor do sistema com os parâmetros estimados.

Table 4.3: Erro quadrático RLC

Erro de I	Erro de V_c	Erro de V_r
1,15574026e-07	8,81899924e-03	8,88219230e-03

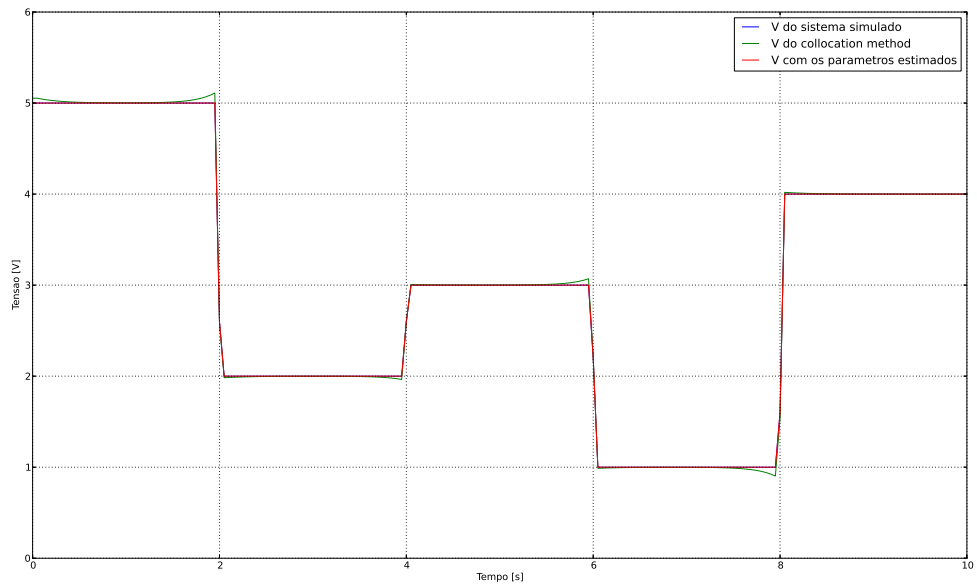


Figure 4.4: Comparação entre as 3 entradas $V(t)$.

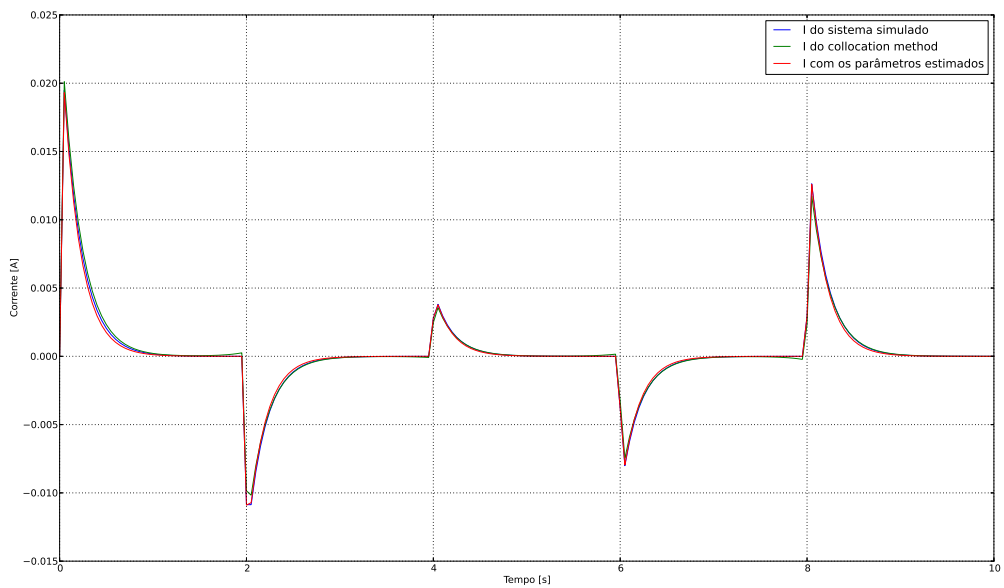


Figure 4.5: Comparação entre as 3 correntes $I(t)$.

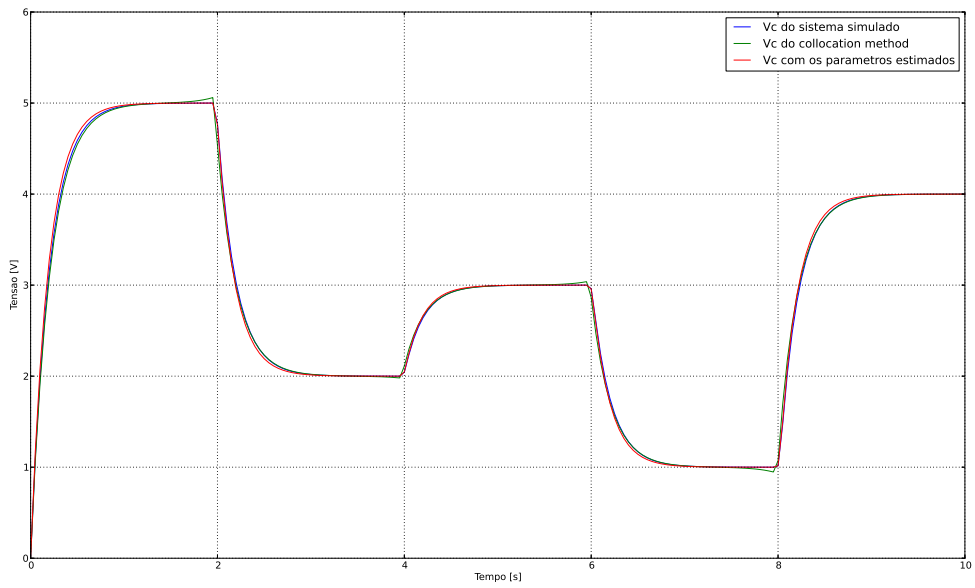


Figure 4.6: Comparação entre as 3 tensões $V_c(t)$.

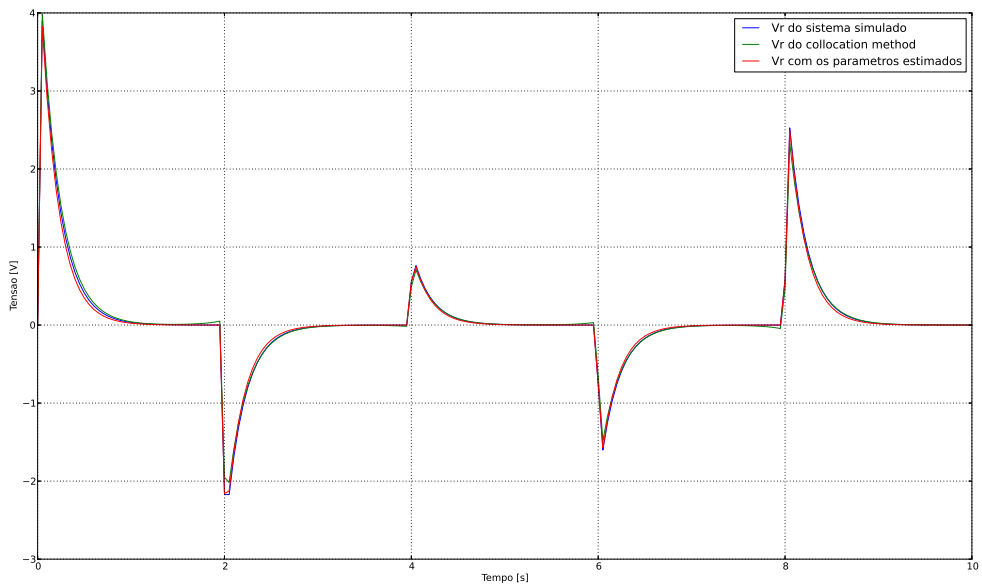


Figure 4.7: Comparação entre as 3 tensões $V_r(t)$.

4.3: Poço Produtor de Petróleo

Os primeiros poços de petróleo nos tempos modernos foram perfurados percussivamente, martelando uma ferramenta a cabo na terra. Pouco depois, ferramentas de cabo foram substituídas por perfuração rotativa, o que poderia perfurar poços a uma profundidade muito maior e em menos tempo. Até os anos 1970, a maioria dos poços de petróleo era vertical, embora imperfeições causassem o desvio, ao menos levemente de uma exata verticalidade da maior parte dos poços (o que levou aos poços direcionais). Entretanto as tecnologias de perfuração direcional modernas permitem poços fortemente desviados, os quais podem, dada suficiente profundidade e com as ferramentas apropriadas, tornarem-se horizontais. Isto é de grande valor, na medida em que rochas reservatório que contém hidrocarbonetos são normalmente horizontais, ou sub-horizontais; um poço horizontal colocado em uma zona de produção tem mais área de superfície na zona de produção que um poço vertical, resultando em uma maior taxa de produção. O uso de perfuração desviada e horizontal também tornou possível chegar a reservatórios a vários quilômetros de distância do local de perfuração (perfuração de alcance estendido), permitindo a produção de hidrocarbonetos localizados abaixo dos locais que são ou difíceis de colocar-se uma sonda de perfuração, ou ambientalmente sensíveis, ou povoados.

O *gas-lift* contínuo, método de elevação do petróleo e gás, baseia-se na injeção contínua de gás a alta pressão na coluna de produção, com o objetivo de gaseificar o fluido desde o ponto de injeção até a superfície. O gás aqui em questão é o gás natural, fruto da produção do próprio poço, que é comprimido em compressores na plataforma e enviado ao annulus do poço através de uma linha de serviço. Na superfície, o controle da injeção de gás no poço é feito através de um regulador de fluxo, ou *choke*. Já no poço há uma válvula de *gas lift* que promove a comunicação entre o annulus e a coluna de produção (tubing) e segue pela linha de produção até chegar à plataforma. A válvula de *gas lift* tem a função de controlar o fluxo de gás e estabelecer contato entre o annulus e o tubing.

Os modelos da rede de extração de petróleo considerados neste trabalho foram modelados em [15]. Esta seção é dedicada ao modelo do poço produtor, onde é aplicado o método proposto para a estimação de parâmetros.

4.3.1: Modelo do Poço

O poço é o principal atuador de um reservatório de petróleo, sendo composto por várias peças distintas: *choke* de *gas-lift*, a válvula de injeção, anulus, tubo de revestimento, e *choke* de produção.

A operação de um poço pode ser descrita pelas seguintes etapas:

1. O gás injetado passa através do *choke* de *gas-lift* e aumenta a pressão do anulus.
2. Como a pressão no anulus excede a pressão na tubulação (*tubing*), o gás começa a fluir através da válvula de injeção (*injection valve*).
3. O gás injetado se mistura com o fluido que emana a partir do reservatório e reduz a densidade geral.
4. Com uma densidade inferior, a contrapressão induzida pelo fluido é reduzida facilitando o fluir para a superfície.

O modelo de Binder [4] é uma extensão do modelo de Eikrem, um modelo que tem sido desenvolvido e usado nos últimos 10 anos [15]. O esquema do poço pode ser observado na Figura 4.8. Uma versão anterior do modelo foi comparado com o simulador OLGA, mesmo com a simplicidade os resultados têm fidelidade considerável. O modelo tem apenas 3 ODEs e algumas equações algébricas e ainda pode representar a maioria das características e restrições do poço. Por estas razões é o modelo usado.

As principais premissas do modelo são:

- A produção de óleo é descrita pela equação de Vogel [8].
- A produção de gás e produção de água é dada pela relação líquido-água (*water-cut*) e razão gás-óleo (GOR).
- A distribuição das massas dos três fluidos ocorre sem demora de transporte.
- A fricção entre os fluidos e a parede do anulo/tubo não é considerada.
- As pressões são obtidas pelas leis de gás ideal e efeito gravitacional de líquidos.

O modelo descreve o sistema dinâmico usando 3 estados e 3 entradas — w_{gl} (vazão do gás de injeção), u_{pc} (*production choke*), e p_{ds} (pressão *downstream*) e suas equações completas são apresentadas no apêndice A.1:

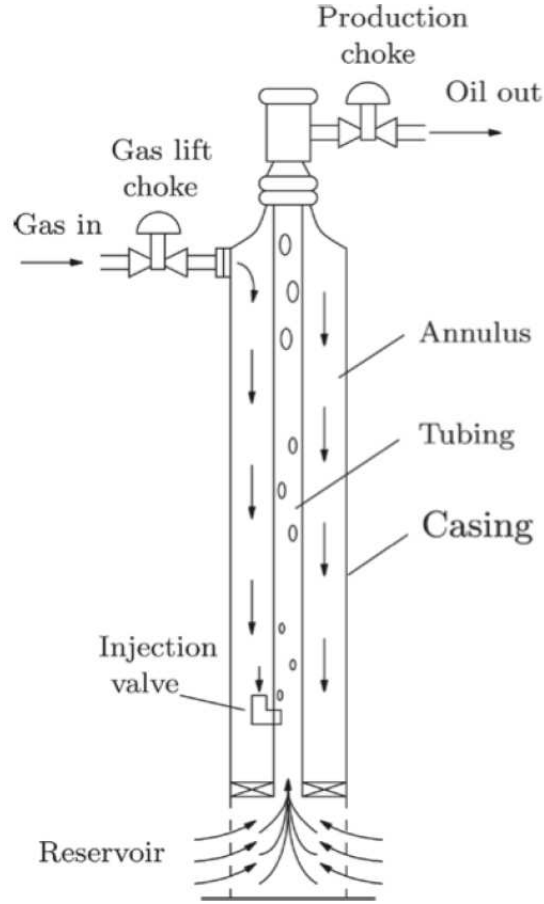


Figure 4.8: Esquema de um Poço.

$$\dot{m} = \begin{bmatrix} \dot{m}_{ga} = w_{in} - w_{gi} \\ \dot{m}_{gt} = w_{gr} + w_{gi} - w_{gp} \\ \dot{m}_{lt} = w_{lr} - w_{lp} \end{bmatrix} \begin{array}{l} \text{Massa de gás no Annulus} \\ \text{Massa de gás na Tubulação} \\ \text{Massa de líquido na Tubulação} \end{array} \quad (4.9)$$

$$w_{out} = f_c(m, p_{ds}, u_{pc}) \quad \text{choke flow}$$

No anulus, a entrada de *gas-lift* w_{in} aumenta a massa de gás m_{ga} do anulus, enquanto o fluxo através da válvula de injeção w_{gi} reduz. O gás proveniente do reservatório w_{gr} adiciona-se ao gás proveniente do anulus w_{gi} e é reduzida a partir do gás que flui através do *choke* de produção w_{gp} descrevendo a dinâmica da massa de gás no tubo. A massa de líquido na tubulação é a integral no tempo do líquido que flui do reservatório w_{lr} descontado pelo fluxo líquido de saída através do *choke* de produção w_{lp} . O vetor de fluxo de gás, óleo e água w_{out} é função dos estados $m = (m_{ga}, m_{gt}, m_{lt})$, da abertura do estrangulamento de produção u_{pc} e da pressão a jusante de estrangulamento p_{ds} , que depende do estado dos oleodutos.

A principal diferença entre os modelos de Binder e Eikrem é que, no modelo de Binder, o reservatório não só produz petróleo e gás, mas água também. Portanto, algumas alterações foram feitas nas equações de densidade e pressão.

Ambos os modelos consideraram a abertura do *choke* de produção u_{pc} como uma variável controlável, no entanto, para reduzir a complexidade do problema se assume que ela está sempre totalmente aberta, uma vez que qualquer outra posição do *choke* iria reduzir a produção de petróleo.

Uma modificação adicional para o modelo de Binder foi necessária para atender a característica de não-retorno no *choke* de *gas-lift*. Deve haver fluxo no *choke* apenas se a pressão no annulus é inferior à pressão antes do *choke*. Devido a esta limitação, a injeção de *gas-lift* deve estar dentro dos limites:

$$p_{gm} \gg p_{ta} \quad (4.10a)$$

$$0 \leq w_{gl} \leq w_{gl}^{\max} = f_{gv}(p_{gm} - p_{ta}) \quad (4.10b)$$

sendo w_{gl}^{\max} o fluxo quando o *choke* de *gas-lift* está totalmente aberto. Esse valor pode ser encontrado utilizando a função do *choke*, que é dependente da diferença de pressão entre o tubo de distribuição de *gas-lift* (p_{gm}) e a parte superior do anulus (p_{ta}).

4.3.2: Estimação dos Parâmetros de um Poço Produtor de Petróleo

Utilizando a mesma metodologia da seção anterior, apresenta-se o problema de estimação dos parâmetros de um poço de petróleo. Os parâmetros a serem estimados são os coeficientes das válvulas de injeção, C_{iv} , e de produção, C_{pc} , e as condições iniciais da massa de gás no anular do poço e na tubulação do poço e massa de líquido na tubulação do poço, respectivamente, m_{ga0} , m_{gt0} e m_{lt0} .

Abaixo encontra-se a Tabela 4.4 com as diferentes situações de simulação, com uma estimativa inicial de $C_{iv} = 0,00017$, $C_{pc} = 0,0019$, $m_{ga0} = 3700$, $m_{gt0} = 1450$ e $m_{lt0} = 3400$

Mesmo com 200 elementos finitos, que apresenta a maior porcentagem de erro, o sistema ainda se comporta como o sistema simulado, como pode ser observado nas Figuras 4.9 e 4.10. A Figura 4.11 mostra a localização das medições dos dados coletados.

Table 4.4: Parâmetros Estimados

Elem fin	T (s)	C_{iv}	C_{pc}	m_{ga0}	m_{gt0}	m_{lt0}
		0,00016	0,0014	3629,07	1389,20	3352,26
200	7,18	0,0001599	0,0014006	3633,32	1388,19	3275,61
500	48,34	0,0001599	0,0014002	3630,80	1389,24	3320,57
1000	99,57	0,0001599	0,0014001	3629,92	1389,26	3336,51

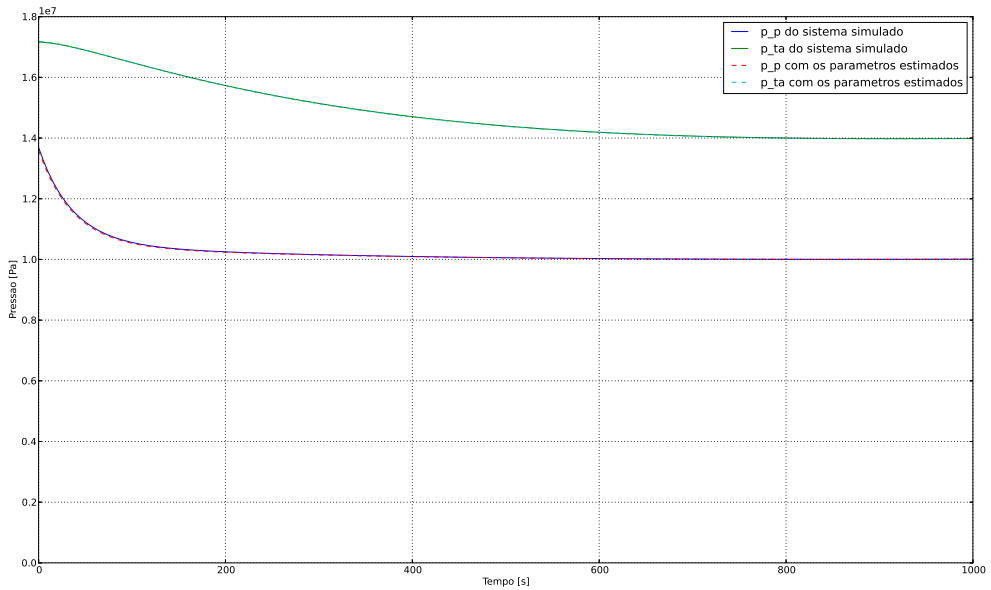


Figure 4.9: Comparação da pressão *upstream* da válvula de produção (p_p) e pressão *downstream* da válvula de injeção (p_{ta}).

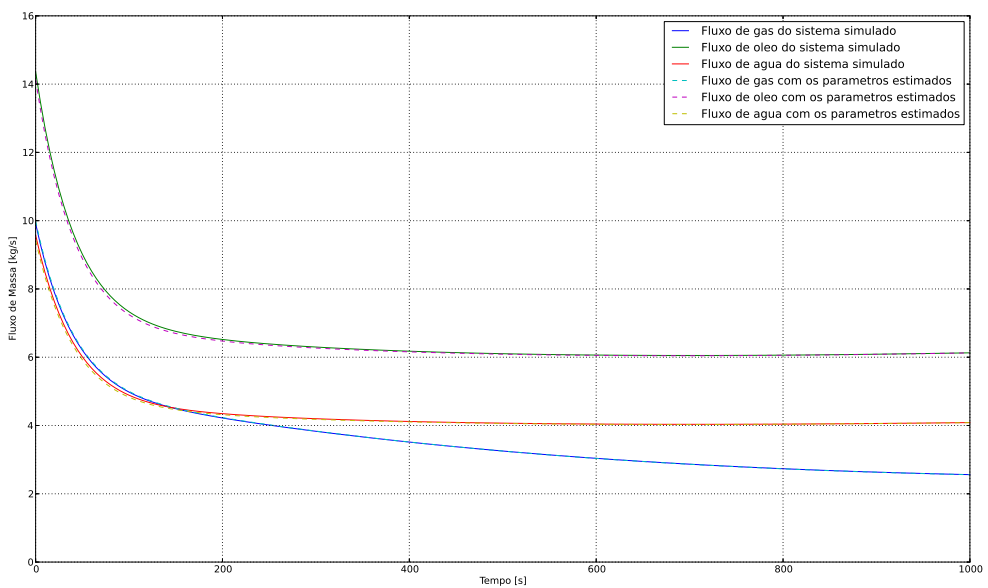


Figure 4.10: Comparação dos fluxos de produção de gás, óleo e água.

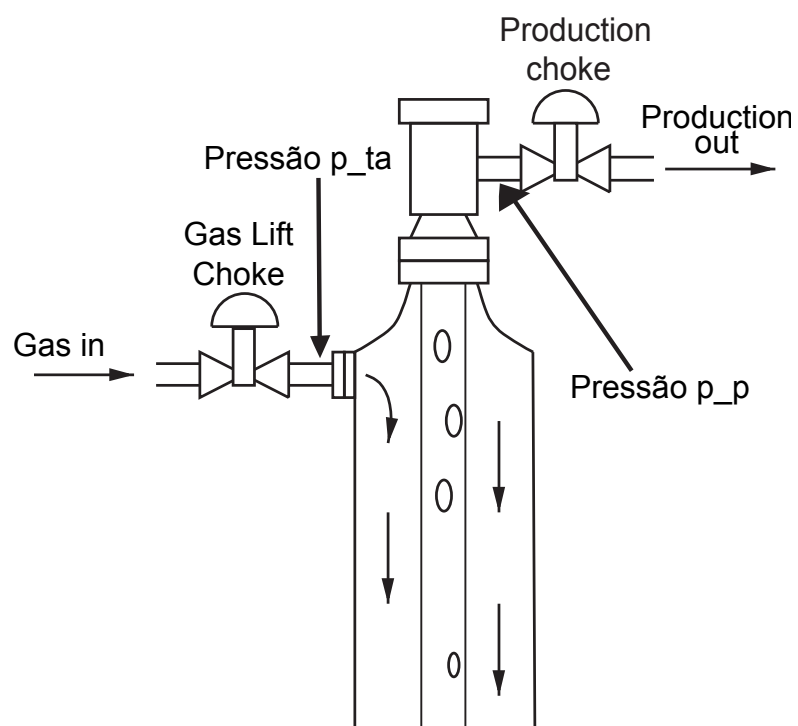


Figure 4.11: Esquema das medições dos dados coletados.

4.4: Estimação do poço com dados

A metodologia empregada até agora utilizava o próprio modelo para estimação, entretanto isso não é possível numa aplicação fora do mundo acadêmico. Pensando nisso, abordou-se um modelo de uma ordem muito maior para se ter como referência para a estimação. Os dados obtidos vêm do modelo descrito em [14].

Após alguns testes, notou-se que o modelo utilizado anteriormente era muito simples e não conseguiria replicar o comportamento dos dados. Por isso foi alterado e decidiu-se por desacoplar o poço, permitindo lidar com o annulus e o *tubing* de maneira separada. Os novos modelos são descritos em [6], um trabalho ainda em andamento e nos Apêndices A.2 e A.3.

4.4.1: Annulus

O modelo modificado do annulus apresenta elementos de atrito para melhor representar os efeitos no sistema real. Essa modificação já é suficiente para aplicar a metodologia proposta. Entretanto o modelo continua simples, com apenas 1 estado, as modificações podem ser observadas no apêndice A.2. Os parâmetros a serem estimados são: o coeficiente da válvula C_{iv} e a condição inicial da massa de gás no anular do poço, m_{ga0} . A Tabela 4.5 apresenta o resultado da estimação. Pode-se perceber, que por causa da separação, mesmo com um número grande de elementos finitos, apresenta um tempo rápido de solução.

Table 4.5: Parâmetros Estimados

Elem fin	T (s)	C_{iv}	m_{ga0}
1440	49,21	0,0001357	6569,80

As Figuras 4.12, 4.13 e 4.14 apresentam as comparações entre as medições e o modelo simulado com os parâmetros apresentados na Tabela 4.5.

4.4.2: Tubing

O modelo modificado do tubing apresenta elementos de atrito para melhor representar os efeitos no sistema real. Algumas modificações adicionais foram necessárias para melhor simular o ambiente real. Um novo estado foi adicionado, a fração mássica de gás e outros parâmetros de *tuning* de atrito. Entretanto o modelo continua simples, com apenas 3 estados, as modificações podem ser observadas no

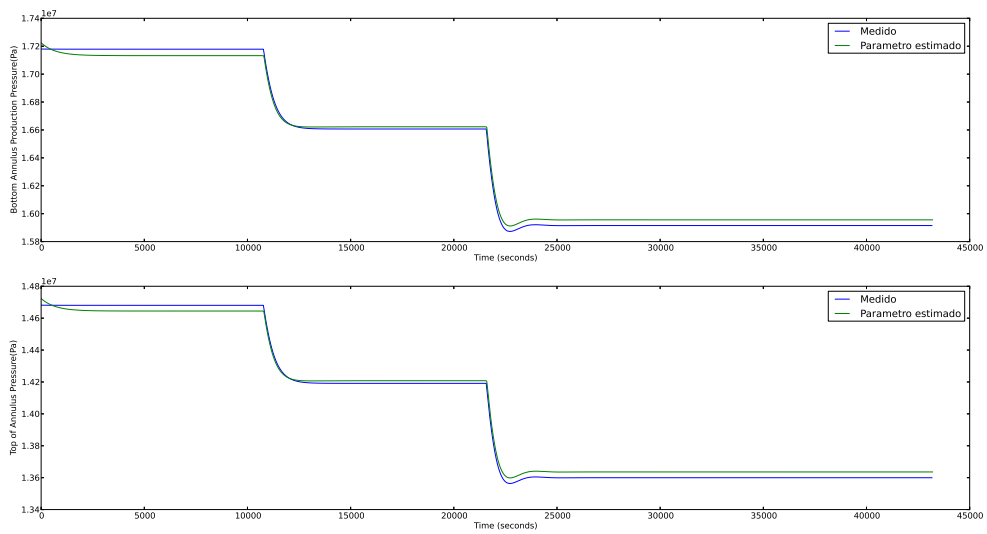


Figure 4.12: Comparação das pressões de fundo e topo do annulus.

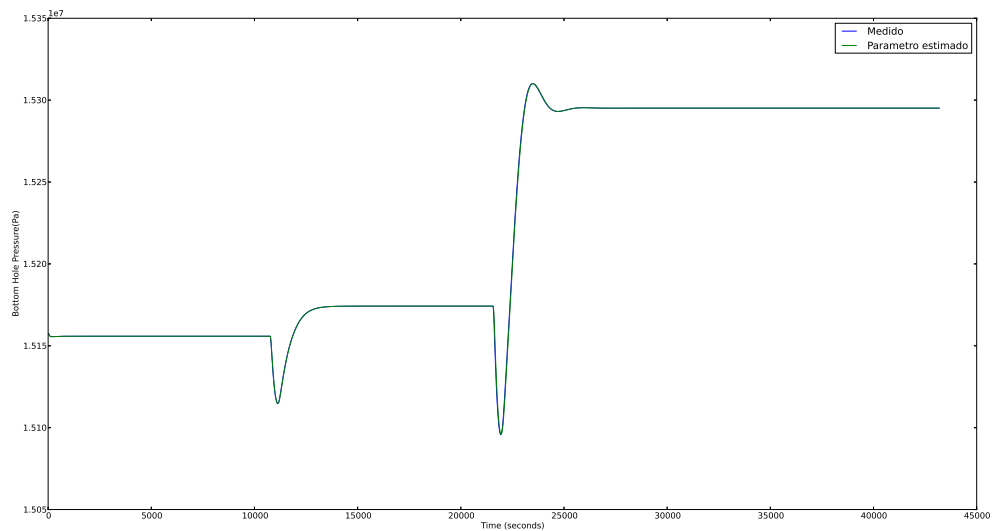


Figure 4.13: Comparação das pressões de *bottom hole*.

apêndice A.3. Os parâmetros a serem estimados são: o coeficiente da válvula C_{pc} , os parâmetros de *tuning* de atrito k e $k2$ e as condições iniciais da massa de gás, de líquido e de fração mássica de gás no tubing do poço, respectivamente, m_{gt0} , m_{lt0} e x_0 . A Tabela 4.6 apresenta o resultado da estimação.

Table 4.6: Parâmetros Estimados

Elem fin	T (s)	C_{pc}	k	$k2$	m_{gt0}	m_{lt0}	x_0
1440	12,84	0,050005	0,464221	1,18581	171,9171	4973,5314	0.14999

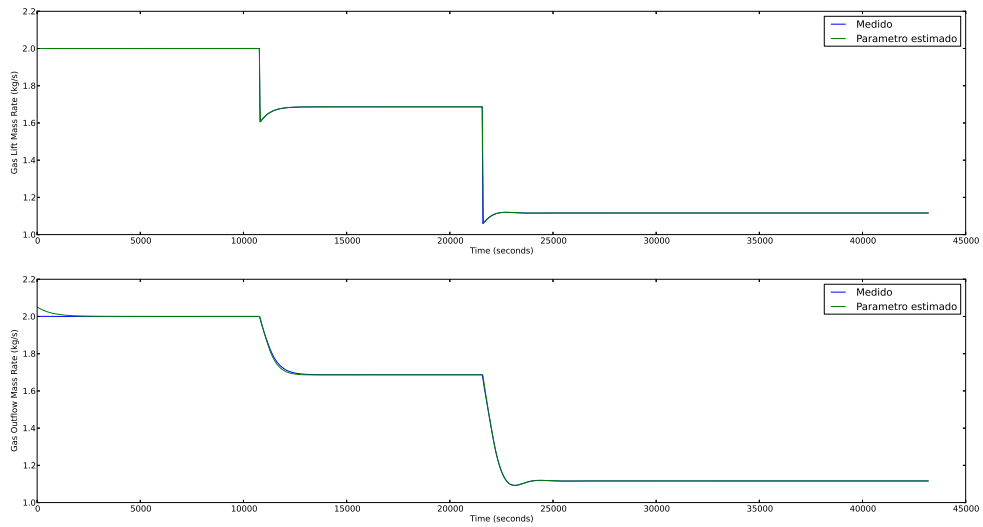


Figure 4.14: Comparação das vazões de entrada e de saída do annulus.

As Figuras 4.15 e 4.16 apresentam as comparações entre as medições e o modelo simulado com os parâmetros apresentados na Tabela 4.6.

Mesmo com as modificações, a pressão de produção (p_p) não consegue seguir a curva medida do simulador. Contudo pode-se observar pelas Figuras 4.15 e 4.16 que as vazões e a pressão de *bottom hole* se aproximam muito do simulador, por isso o resultado ainda é satisfatório, pois ainda pode-se utilizar disso para algumas aplicações.

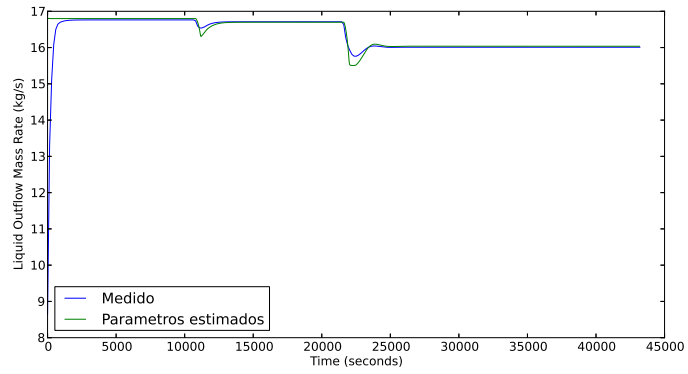
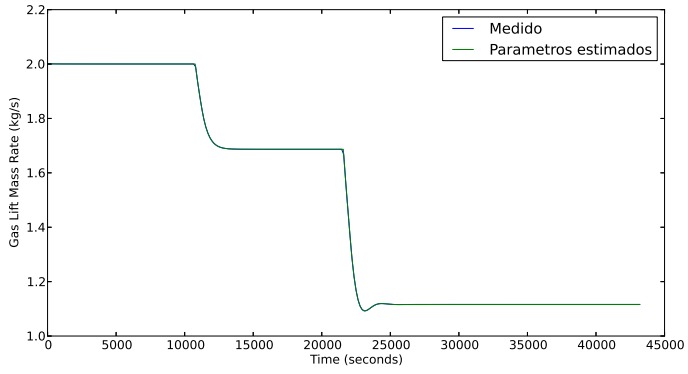
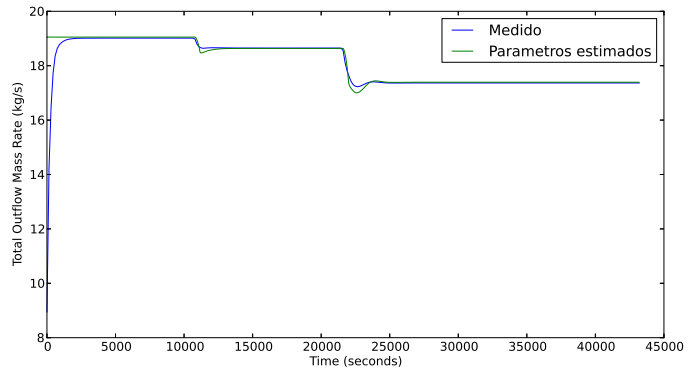
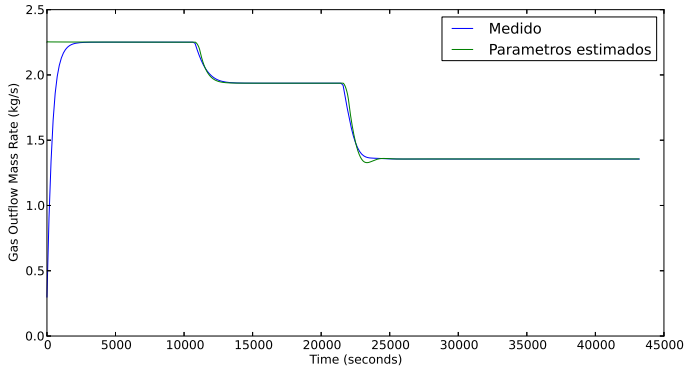


Figure 4.15: Comparação das vazões de entrada e saída.

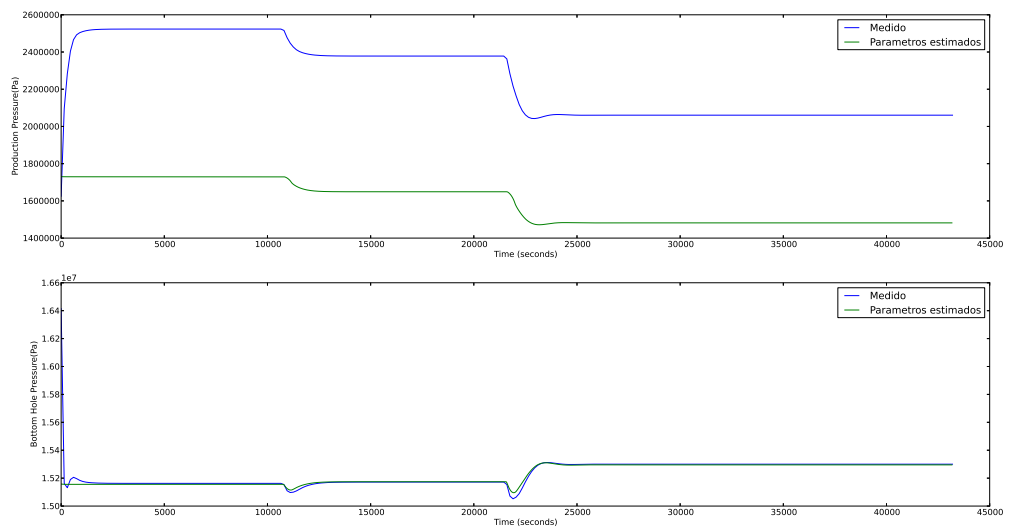


Figure 4.16: Comparação das pressões de produção e de *bottom hole*.

Chapter 5: Conclusão

Este trabalho apresentou o *collocation method*, a diferença entre o Gauss *collocation* e Radau *collocation*, expondo seu equacionamento e dedução matemática. Um exemplo de Radau e Gauss *collocation* é demonstrado para melhor fixar o método.

Uma ferramenta interessante foi apresentada neste trabalho, o JModelica.org que implementa os compiladores para Modelica e Optimica e que reúne ferramentas para simulação e otimização. Modelica e Optimica permitem modelar e criar problemas de otimização dinâmica com facilidade. A plataforma JModelica.org teve um papel importante no desenvolvimento e análise dos modelos e da estrutura de controle. A linguagem Modelica ajudou a especificar DAEs modulares que representassem componentes físicos da rede de produção, enquanto a linguagem Optimica nos permitiu definir estratégias de estimação.

A estimação foi implementada utilizando o *collocation method* combinado com técnicas de otimização que são mais robustas e viáveis do que outros métodos (e.g; filtro de Kalman). O método foi demonstrado, seguido por um exemplo explicativo e demonstração de como ele pode ser usado para formular problemas de otimização não-linear.

A validação da estratégia desenvolvida com o circuito RLC, permitiu o avanço do estudo para um modelo de poço produtor de gás e óleo de pequena ordem. Com os resultados apresentados abordou-se um problema mais prático. Utilizando dados de um simulador de grande ordem foi possível estimar parâmetros que aproximavam o modelo de menor ordem ao do simulador. Este resultado permite a utilização de simuladores mais simples, facilitando cálculos (e.g; controle) e diminuindo o tempo de processamento para simulações. Como continuação deste trabalho, pode-se prosseguir na verificação dos modelos de forma a melhorá-los, mas mantendo a sua forma simples, a ponto de se aproximar ainda mais do simulador. Outro ponto seria a utilização dessa mesma estratégia em outros componentes de uma rede de extração de óleo e gás.

Bibliography

- [1] Johan Åkesson. Optimica: An extension of modelica supporting dynamic optimization. Proc. 6th Int. Model. Conf. 2008, 2008.
- [2] JS Albuquerque and LT Biegler. Decomposition algorithms for on-line estimation with nonlinear models. Comput. Chem. Eng., 1995.
- [3] UM Ascher and LR Petzold. Computer methods for ordinary differential equations and differential algebraic equations. SIAM: Society for Industrial and Applied Mathematics, 1998.
- [4] BJT Binder. Production Optimization in a Cluster of Gas-Lift Wells, 2012.
- [5] GO Brown. The history of the Darcy-Weisbach equation for pipe flow resistance. Environ. Water Resour. Hist., pages 34–43, 2002.
- [6] Marco Aurelio S. de Aguiar, Andrés Cudas, and Eduardo Camponogara. A Collocation-Based Approach for Optimal Control of Offshore Oil Production Networks.
- [7] Carlos Henrique Farias dos Santos. Teoria E Pratica Do Filtro De Kalman.
- [8] K.A. Fattah, M Elias, HA El-Banbi, and ESA El-Tayeb. New Inflow Performance Relationship For Solution-Gas Drive Oil Reservoirs. J. Pet. Technol., 20(1):343–366, 2012.
- [9] Shivakumar Kameswaran and Lorenz T. Biegler. Convergence rates for direct transcription of optimal control problems using collocation at Radau points. Comput. Optim. Appl., 41(1):81–126, November 2007.
- [10] S Karneswaran and LT Biegler. Convergence rates for direct transcription of optimal control problems with final-time equality constraints using collocation at Radau points. Am. Control Conf. . . ., pages 165–171, 2006.
- [11] Daniel Vitor De Lucena. Filtering and improving of spectra with Kalman filter. Abakós, 1(2), 2013.
- [12] Fredrik Magnusson. Collocation methods in JModelica. org, 2012.

- [13] Bethany Nicholson, Rodrigo López-Negrete, and Lorenz T. Biegler. On-line state estimation of nonlinear dynamic systems with gross errors. Comput. Chem. Eng., December 2013.
- [14] Agostinho Plucenio, C. A. Ganzaroli, and Daniel J. Pagano. Stabilizing gas-lift well dynamics with free operating point. In 2012 IFAC Work. Autom. Control Offshore Oil Gas Prod., number 2010, pages 95–100, 2012.
- [15] Marco Schmitz de Aguiar. Optimal oil production network control using Modelica, 2013.
- [16] Greg Welch and Gary Bishop. An introduction to the Kalman filter, 1995.

Appendix A : Modelos Dinâmicos

A.1: Modelo do Poço

Estas são as equações do modelo descrito em 4.3.

Balanco de Massa:

$$\dot{m}_{ga} = w_{gl} - w_{gi} \quad (\text{A.1a})$$

$$\dot{m}_{gt} = w_{gr} + w_{gi} - w_{gp} \quad (\text{A.1b})$$

$$\dot{m}_{lt} = w_{gl} - w_{gi} \quad (\text{A.1c})$$

Fluxo de Massa:

$$w_{gi} = C_{iv} \sqrt{\rho_{gi} \max(0, p_{ai} - p_{ti})} \quad (\text{A.2a})$$

$$w_p = C_{pc} \sqrt{\rho_p \max(0, p_{ai} - p_{ti})} \quad (\text{A.2b})$$

$$w_{gp} = \frac{m_{gt}}{m_t} w_p \quad (\text{A.2c})$$

$$w_{lp} = \frac{m_{lt}}{m_t} w_p \quad (\text{A.2d})$$

$$w_{op} = (1 - r_{wc}) w_{lp} \quad (\text{A.2e})$$

$$w_{wp} = r_{wc} w_{lp} \quad (\text{A.2f})$$

$$w_{lr} = \rho_l Q_{\max} \left(1 - (1 - C) \left(\frac{p_{bh}}{p_r} \right) - C \left(\frac{p_{bh}}{p_r} \right)^2 \right) \quad (\text{A.2g})$$

$$w_{gr} = r_{glr} w_{lr} \quad (\text{A.2h})$$

$$w_{gl}^{\max} = C_{gl} \sqrt{\rho_{gl} \max(0, p_{gm} - p_{ta})} \quad (\text{A.2i})$$

Pressões

$$p_{ai} = \left(\frac{RT_a}{V_a M_g} + \frac{g}{2A_a} \right) m_{ga} \quad (\text{A.3a})$$

$$p_p = \frac{RT_t m_{gt}}{M_g V_t - M_g \rho_l^{-1} m_{lt}} - \frac{gm_t}{2A_t} \quad (\text{A.3b})$$

$$p_{ti} = p_p + \frac{gm_t}{A_t} \quad (\text{A.3c})$$

$$p_{bh} = \frac{\left(1 + r_{glr} + \frac{r_{glr} g M_g L_w}{2RT_t} \right) p_{ti} + \rho_l g L_w}{1 + r_{glr} - \frac{r_{glr} g M_g L_w}{2RT_t}} \quad (\text{A.3d})$$

$$p_{ta} = \left(\frac{RT_a}{V_a M_g} - \frac{g}{2A_a} \right) m_{ga} \quad (\text{A.3e})$$

Densidades:

$$\rho_{gi} = \frac{M_g}{RT_a} p_{ai} \quad (\text{A.4a})$$

$$\rho_p = \frac{\rho_l M_g p_p m_t}{\rho_l RT_t m_{lt} + M_g p_p m_{gt}} \quad (\text{A.4b})$$

$$\rho_l = r_{wc} \rho_w + (1 - r_{wc}) \rho_o \quad (\text{A.4c})$$

$$\rho_{gl} = \frac{M_g}{RT_a} p_{gm} \quad (\text{A.4d})$$

Table A.1: Variáveis do modelo do poço

Nome	Tipo	Explicação
\dot{m}_{ga}	Estado	Massa de gás no annulus
\dot{m}_{gt}	Estado	Massa de gás no tubing
\dot{m}_{lr}	Estado	Massa de líquido no the tubing
w_{gi}	Algébrico	Vazão de massa na válvula de injeção
w_p	Algébrico	Vazão de massa no choke de produção
w_{gp}	Algébrico	Produção de gás
w_{lp}	Algébrico	Produção de líquido
w_{op}	Algébrico	Produção de óleo
w_{wp}	Algébrico	Produção de água
w_{lr}	Algébrico	Líquido extraído do reservatório
w_{gr}	Algébrico	Gás extraído do reservatório
ρ_{gi}	Algébrico	Densidade do gás upstream da válvula de injeção
ρ_p	Algébrico	Densidade da produção upstream do choke de produção
p_{ai}	Algébrico	Pressão upstream da válvula de injeção
p_p	Algébrico	Pressão upstream da válvula de produção
p_{ti}	Algébrico	Pressão downstream de injeção
p_{bh}	Algébrico	Pressão no bottom hole

Table A.2: Variáveis do modelo do poço

Nome	Tipo	Explicação
C_{iv}	Tunning	Coefficiente de válvula de injeção
C_{pc}	Tunning	<i>Choke</i> de produção
r_{wc}	Reservatório	<i>Watercut</i> do reservatório
Q_{max}	Reservatório	Vazão teórica máxima do reservatório
C	Constante	Constante de Voegel
p_r	Reservatório	Pressão de reservatório
r_{gor}	Reservatório	Razão gás-óleo
R	Constante	Constante de gás ideal
T_a	Poço	Temperatura do annulus
g	Constante	Gravidade
V_a	Poço	Volume do Annulus
M_g	Parâmetro	Massa molar do gás
A_a	Poço	Área da secção do annulus
T_t	Poço	Temperatura do tubing
V_t	Poço	Volume do tubing
A_t	Poço	Área da secção do tubing
r_{glr}	Reservatório	Razão gás-líquido
L_w	Poço	Comprimento do tubing do reservatório para ponto de injeção
ρ_l	Parâmetro	Densidade de massa líquida
ρ_w	Parâmetro	Densidade de massa de água
ρ_o	Parâmetro	Densidade de massa de óleo

A.2: Annulus

A equação Darcy-Weisbach [5] é dada por:

$$p_f - p_0 = f_D \frac{\rho L U^2}{2D} \quad (\text{A.5})$$

onde p_f e p_0 são pressões em dois pontos diferentes em um tubo, f_D é fator de atrito de Darcy, L é a distância entre os pontos, D o diâmetro do tubo, e U é a velocidade média (m/s) do fluido. O f_D é dado por:

$$f_D = \left(-1.8 \log_{10} \left[\left(\frac{\varepsilon}{3.7D} \right)^{1.11} + \frac{6.9}{Re} \right] \right)^{-2} \quad (\text{A.6})$$

porque a velocidade média é difícil de calcular e fórmulas complexas podem resultar em equações implícitas, que são caras para resolver, é aproximada por

$$U \approx \frac{w_{in}}{A\bar{\rho}} = \frac{w_{in}V}{Am} = \frac{w_{in}L}{m} \quad (\text{A.7})$$

com A sendo a área de seção transversal, $\bar{\rho}$ a densidade média do fluido que pode ser substituído por m/V , onde m é a massa de líquido na seção e V o volume. O número de Reynold Re pode ser definido como::

$$Re = \frac{\rho DU}{\mu} \approx \frac{w_{in}D}{\mu A} \quad (\text{A.8})$$

onde μ é a viscosidade dinâmica e w_{in} é o fluxo de entrada no annulus.

Para um comprimento substancialmente pequeno dh a queda de pressão $d\Delta p$ é:

$$dp = f_D \frac{1}{D} \frac{\rho U^2}{2} dh \quad (\text{A.9})$$

Usando a fórmula de pressão hidrostática:

$$dp = -\rho g dh \quad (\text{A.10})$$

Somando-se a dinâmica de ambos os fenômenos, a resultante é

$$dp = \rho \left[-g + f_D \frac{1}{D} \frac{U^2}{2} \right] dh \quad (\text{A.11})$$

se um fluxo contrário na direção de h é assumida. Usando a lei dos gases ideais::

$$pV = nRT \Rightarrow pV = \frac{m}{M} RT \Rightarrow \rho = \frac{pM}{RT} \quad (\text{A.12a})$$

que produz:

$$dp = \frac{pM}{RT} \left[-g + f_D \frac{1}{D} \frac{U^2}{2} \right] dh \Rightarrow \quad (\text{A.13})$$

$$\frac{dp}{p} = \frac{M}{RT} \left[-g - f_D \frac{1}{D} \frac{U^2}{2} \right] dh \quad (\text{A.14})$$

por uma questão de legibilidade C é definido como:

$$C = \frac{M}{RT} \left[-g - f_D \frac{1}{D} \frac{U^2}{2} \right] \quad (\text{A.15})$$

integrando ambos os lados:

$$\log p_f - \log p_0 = C(h_f - h_0) \quad (\text{A.16a})$$

$$p_f = p_0 e^{C(h_f - h_0)} \quad (\text{A.16b})$$

Aplicando ao annulus:

$$p_{ta} = p_{ba} e^{CL_a} \quad (\text{A.17})$$

onde ba significa fundo do annulus, e ta significa topo do annulus, e $h_{ta} - h_{ba}$ é igual ao comprimento do annulus L_a . A variável C é dada por:

$$C = \frac{M_g}{RT_a} \left[-g + f_D \frac{1}{D_a} \frac{U_{ga}^2}{2} \right] \quad (\text{A.18})$$

A pressão no topo do annulus também pode ser definida como:

$$p_{ta} = p_{ba} - g\bar{\rho}L_a + f_D \frac{L_a}{D} \frac{\bar{\rho}U_{ga}^2}{2} \quad (\text{A.19})$$

A densidade média no annulus é dado por:

$$\bar{\rho} = \frac{m_{ga}}{V_a} \quad (\text{A.20})$$

substituição na equação:

$$p_{ta} = p_{ba} - \frac{gm_{ga}}{A_a} + f_D \frac{L_a}{D_a} \frac{m_{ga}U_{ga}^2}{2V_a} \quad (\text{A.21})$$

O que resulta em um sistema não linear com duas equações e duas variáveis, com a

solução analítica:

$$p_{ba} - \frac{gm_{ga}}{A_a} + f_D \frac{L_a}{D_a} \frac{m_{ga} U_{ga}^2}{2V_a} = p_{ba} e^{CL_a} \quad (\text{A.22a})$$

$$p_{ba} (1 - e^{CL_a}) = \frac{gm_{ga}}{A_a} - f_D \frac{L_a}{D_a} \frac{m_{ga} U_{ga}^2}{2V_a} \quad (\text{A.22b})$$

$$p_{ba} = \frac{\frac{gm_{ga}}{A_a} - f_D \frac{L_a}{D_a} \frac{m_{ga} U_{ga}^2}{2V_a}}{(1 - e^{CL_a})} \quad (\text{A.22c})$$

$$p_{ta} = p_{ba} - \frac{gm_{ga}}{A_a} + f_D \frac{m_{ga} U_{ga}^2}{2D_a A_a} \quad (\text{A.22d})$$

O conjunto de equações é:

$$U_{ga} = \frac{w_{in} L_a}{m_{ga}} \quad (\text{A.23a})$$

$$Re = \frac{w_{in} D_a}{\mu_g A_a} \quad (\text{A.23b})$$

$$f_D = \left(-1.8 \log_{10} \left[\left(\frac{\varepsilon_a}{3.7 D_a} \right)^{1.11} + \frac{6.9}{Re} \right] \right)^{-2} \quad (\text{A.23c})$$

$$C = \frac{M_g}{RT_a} \left[-g + f_D \frac{1}{D_a} \frac{U_{ga}^2}{2} \right] \quad (\text{A.23d})$$

$$p_{ba} = \frac{\frac{gm_{ga}}{A_a} - f_D \frac{L_a}{D_a} \frac{m_{ga} U_{ga}^2}{2V_a}}{(1 - e^{CL_a})} \quad (\text{A.23e})$$

$$p_{ta} = p_{ba} - \frac{gm_{ga}}{A_a} + f_D \frac{m_{ga} U_{ga}^2}{2D_a A_a} \quad (\text{A.23f})$$

$$\rho_{gi} = \frac{p_{ba} M_g}{RT_a} \quad (\text{A.23g})$$

A.3: Tubing II

A equação Darcy-Weisbach [5] é dada por:

$$p_f - p_0 = f_D \frac{\rho L U^2}{2D} \quad (\text{A.24})$$

Para um comprimento substancialmente pequeno dh a queda de pressão dp é:

$$dp = f_D \frac{1}{D} \frac{\rho U^2}{2} dh \quad (\text{A.25})$$

onde p_f e p_0 são pressões em dois pontos diferentes em um tubo, f_D é fator de atrito de Darcy, L é a distância entre os pontos, D o diâmetro do tubo, e U é a velocidade

média (m/s) do fluido. O f_D é dado por:

$$f_D = k \left(-1.8 \log_{10} \left[\left(\frac{\varepsilon}{3.7D} \right)^{1.11} + \frac{6.9}{Re} \right] \right)^{-2} \quad (\text{A.26})$$

porque a velocidade média é difícil de calcular e fórmulas complexas podem resultar em equações implícitas, que são caras para resolver, é aproximada por

$$U_m = U_{s,g} + U_{s,l} = \frac{w_l}{A_l \rho_l} + \frac{w_g}{A_g \rho_g} \quad (\text{A.27})$$

$$\approx \frac{(w_{l,in} + w_{l,out})L}{2m_{lt}} + \frac{(w_{g,in} + w_{g,out})L}{2m_{gt}} \quad (\text{A.28})$$

e

$$Re = \frac{\rho D U_m}{\mu} \approx \frac{(m_{lt} + m_{gt}) D U_m}{V \mu A} \quad (\text{A.29})$$

Sendo a fração de massa de gás definido como:

$$x = \frac{\alpha \rho_g}{\alpha \rho_g + (1 - \alpha) \rho_l} \quad (\text{A.30})$$

e a fração de vácuo definida como:

$$\alpha = \frac{U_{s,g}}{C_0(U_{s,g} + U_{s,l}) + V_d} \quad (\text{A.31})$$

substituindo um no outro:

$$x = \frac{q_g}{q_g + q_l + V_d A \rho_l} \quad (\text{A.32})$$

e isolando α na primeira equação:

$$\alpha = \frac{x \rho_l}{(1 - x) \rho_g + x \rho_l} \quad (\text{A.33})$$

A mistura de fluidos médio num determinado ponto é:

$$\rho_m = \alpha \rho_g + (1 - \alpha) \rho_l \quad (\text{A.34})$$

substituindo na primeira equação temos:

$$\rho_m = \frac{\alpha \rho_g}{x} \quad (\text{A.35})$$

com

$$\rho_g = \frac{p M_g}{RT} \quad (\text{A.36})$$

a equação é expandido para:

$$\rho_m = \frac{x\rho_l}{(1-x)\frac{pM_g}{RT} + x\rho_l} \frac{1}{x} \frac{pM_g}{RT} \quad (\text{A.37a})$$

$$= \frac{pM_g\rho_l}{(1-x)pM_g + x\rho_l RT} \quad (\text{A.37b})$$

Sabendo que a equação hidrostática é:

$$dp = -\rho_m g dh \quad (\text{A.38a})$$

se somarmos os dois fenômenos, temos:

$$dp = \rho_m \left[-g - f_D \frac{\rho U_m^2}{2D} \right] dh \quad (\text{A.39a})$$

$$= \frac{pM_g\rho_l}{(1-x)pM_g + x\rho_l RT} \left[-g - f_D \frac{\rho U_m^2}{2D} \right] dh \quad (\text{A.39b})$$

por uma questão de legibilidade C é definido como:

$$C = M_g\rho_l \left[-k_2 g - f_D \frac{1}{D} \frac{U^2}{2} \right] \quad (\text{A.40})$$

consequentemente,

$$dp = \frac{p}{(1-x)M_g p + x\rho_l RT} C dh \quad (\text{A.41a})$$

$$\frac{(1-x)M_g p + x\rho_l RT}{p} dp = C dh \quad (\text{A.41b})$$

que tem como solução:

$$\left[(1-x)M_g p + x\rho_l RT \log p \right] \Big|_{p=p_{ip}}^{p_{wh}} = [Ch] \Big|_{h=h_{ip}}^{h_{wh}} \quad (\text{A.42a})$$

assumindo $L_t = h_{wh} - h_{ip}$

$$\left[(1-x)M_g(p_{wh} - p_{ip}) + x\rho_l RT(\log p_{wh} - \log p_{ip}) \right] = CL_t \quad (\text{A.43})$$

além disso, sabemos que

$$p_{wh} = p_{ip} - \Delta p_{t,m} - \Delta p_{t,U} \quad (\text{A.44})$$

$$\Delta p_{t,m} = \frac{g(m_{gt} + m_{lt})}{A_t} \quad (\text{A.45})$$

$$\Delta p_{t,U} = f_D \frac{\bar{\rho}_m L U^2}{D} \frac{1}{2} = f_D \frac{(m_{gt} + m_{lt}) U^2}{2A_t D} \quad (\text{A.46})$$

com $\bar{\rho}_m = \frac{m_{gt} + m_{lt}}{V_t}$, pode ser substituído na (A.43)

$$x\rho_l RT(\log p_{wh} - \log p_{ip}) = CL_t - (1-x)M_g(p_{wh} - p_{ip}) \quad (\text{A.47a})$$

$$(\log p_{wh} - \log p_{ip}) = \frac{CL_t + (1-x)M_g(\Delta p_{t,m} + \Delta p_{t,U})}{x\rho_l RT} \quad (\text{A.47b})$$

$$p_{wh} = p_{ip} e^{\frac{CL_t + (1-x)M_g(\Delta p_{t,m} + \Delta p_{t,U})}{x\rho_l RT}} \quad (\text{A.47c})$$

Resolvendo analiticamente:

$$p_{ip} = \frac{\Delta p_{t,m} + \Delta p_{t,U}}{1 - e^{\frac{CL_t + (1-x)M_g(\Delta p_{t,m} + \Delta p_{t,U})}{x\rho_l RT}}} \quad (\text{A.48a})$$

$$p_{wh} = p_{ip} - \Delta p_{t,m} - \Delta p_{t,U} \quad (\text{A.48b})$$

$$\Delta p_{t,m} = \frac{g(m_{gt} + m_{lt})}{A_t} \quad (\text{A.48c})$$

$$\Delta p_{t,U} = f_D \frac{\bar{\rho}_m L U^2}{D} \frac{1}{2} = f_D \frac{(m_{gt} + m_{lt}) U^2}{2A_t D} \quad (\text{A.48d})$$

Appendix B: Modelos de Simulação

Este apêndice apresenta os códigos utilizados para as simulações comentadas no Capítulo 4.

B.1: Modelo RLC

```
model RLC // Valores iniciais e parâmetros.  
parameter Real  $x1_0 = 0$ ;  
parameter Real  $x2_0 = 0$ ;  
parameter Real R(min = 1, max = 500, nominal = 100) = 200;  
parameter Real L(min = 1e-4, max = 0.1, nominal = 1e-2) = 0.01;  
parameter Real C(min = 1e-7, max = 0.01, nominal = 1e-3) = 1e-3;  
Real Vr;  
  
// Estados  
Real Vc(start =  $x1_0$ , fixed = true, nominal = 1);  
Real I(start =  $x2_0$ , fixed = true, nominal = 1e-5);  
  
// Sinal de Controle  
input Real V(start = 0, fixed = true);  
  
equation  
der(Vc) = I/C;  
der(I) = V/L - Vc/L - R*I/L;  
Vr = R*I;  
end RLC;
```

B.2: Modelo de Otimização do RLC

```
optimization RLCParEst (startTime = 0, finalTime = 10)
```

```

extends RLC(R(free=true), L(free=true), C(free=true) );
end RLCParEst;

```

B.3: Código Python para Simulação e Estimação

```

// Importa funções para compilar
from pymodelica import compile_fmu
from pyfmi import load_fmu
import numpy as N
from pyjmi import transfer_optimization_problem
from collections import OrderedDict
from pyjmi.optimization.casadi_collocation import MeasurementData

// SIMUALTION OF THE RLC FOR DATA SAMPLING

// Compila o modelo
simulationModelName = compile_fmu("RLC","RLC.mo")

// Carrega o modelo
simulationModel = load_fmu(simulationModelName)

t = N.linspace(0.,10.,200) // Criar pontos uniformemente espaçados
V5 = 5*N.ones(40) // Cria o vetor de entrada
V2 = 2*N.ones(40) // Cria o vetor de entrada
V3 = 3*N.ones(40) // Cria o vetor de entrada
V1 = 1*N.ones(40) // Cria o vetor de entrada
V4 = 4*N.ones(40) // Cria o vetor de entrada
V = N.concatenate((V5,V2,V3,V1,V4),axis = 0)

V_traj = N.transpose(N.vstack((t,V))) // Criar a matriz de dados e transpõe a
matriz para a forma correta

input_object = ('V', V_traj)

```

```

simulationOpts = simulationModel.simulate_options() // Recupera as opções pa-
drão

simulationOpts['ncp'] = 200 // Altera o número de pontos de comunicação.

simulationOpts['CVode_options']['maxh'] = 0.05

//simula o sistema RLC

res1 = simulationModel.simulate(final_time=2, options = simulationOpts, input=
input_object)

// Obter dados amostrados a partir dos resultados simulados

Vc_sim = res1['Vc']

I_sim = res1['I']

t_sim = res1['time']

V_sim = res1['V']

Vr_sim = res1['Vr']

// SIMULAÇÃO PARA PRIMEIRA SUPOSIÇÃO DO ALGORITMO DE OTIMIZA-
ÇÃO

// Reseta o modelo

simulationModel.reset()

// Suposição inicial

simulationModel.set('R',400)

simulationModel.set('L',0.02)

simulationModel.set('C',2e-3)

// simula o sistema RLC

res2 = simulationModel.simulate(final_time=2, options = simulationOpts, input=
input_object)

// PROBLEMA DE ESTIMAÇÃO DE PARÂMETROS

// Compila o modelo

optimizationModel = transfer_optimization_problem("RLCParEst",

```

```

["RLCParEst.mop", "RLC.mo"])
Q = N.diag([1., 1., 1., 1.])
data_V = N.vstack([t_sim, V_sim])
data_Vc = N.vstack([t_sim, Vc_sim])
data_Vr = N.vstack([t_sim, Vr_sim])
data_I = N.vstack([t_sim, I_sim])
unconstrained = OrderedDict()
constrained = OrderedDict()
constrained['V'] = data_V
unconstrained['Vc'] = data_Vc
unconstrained['Vr'] = data_Vr
unconstrained['I'] = data_I
measurement_data = MeasurementData(Q=Q, unconstrained=unconstrained,
constrained=constrained)
n_e = 200 // Número de elementos finitos no algoritmo de collocation
// Obtem um objeto de opções para o algoritmo de otimização
opt_opts = optimizationModel.optimize_options()
// Defina a trajetória e número de collocation points
opt_opts['n_e'] = n_e
opt_opts['init_traj'] = res2.result_data
opt_opts['n_cp'] = 1
opt_opts['measurement_data'] = measurement_data
// Resolve o problema de otimização
res3 = optimizationModel.optimize(options=opt_opts)
// Extração de valores ótimos de parâmetros
R_opt = res3.final("R")

```

```

L_opt = res3.final("L")
C_opt = res3.final("C")

// Imprima os valores dos parâmetros ideais
print 'R: ' + str(R_opt)
print 'L: ' + str(L_opt)
print 'C: ' + str(C_opt)

// Reseta o modelo
simulationModel.reset()

// Estabelece os parâmetros encontrados para simulação
simulationModel.set('R',R_opt)
simulationModel.set('L',L_opt)
simulationModel.set('C',C_opt)

// simula o sistema RLC
res4 = simulationModel.simulate(final_time=2, options = simulationOpts, input=
input_object)

// Compila o modelo
simulationModelName = compile_fmu("RLC2","RLC2.mo")

// Carrega o modelo
simulationModel = load_fmu(simulationModelName)

//simula o sistema RLC
res5 = simulationModel.simulate(final_time=10, options = simulationOpts, input=
input_object)

```